(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2005-128733
(P2005-128733A)
(43)公開日　平成17年5月19日(2005.5.19)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 12/00	G06F 12/00	545B	5B065
G06F 3/06	G06F 12/00	514E	5B082
	G06F 3/06	301Z	
	G06F 3/06	540	

審査請求 未請求　請求項の数 17　OL　（全 16 頁）

(21)出願番号　特願2003-362750 (P2003-362750)
(22)出願日　平成15年10月23日 (2003.10.23)

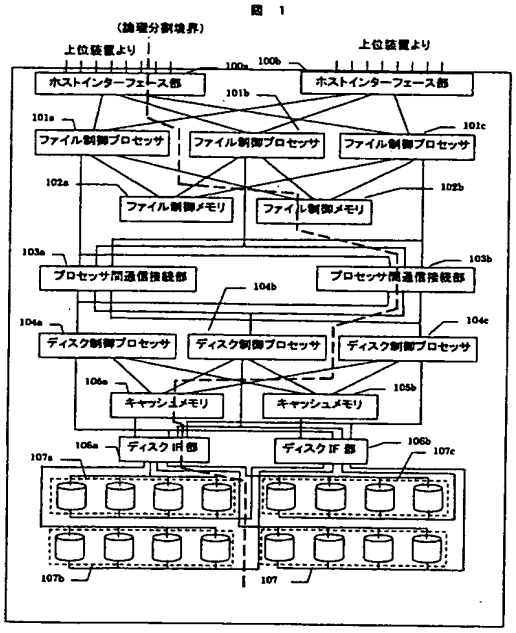
(71)出願人　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　100075096
弁理士　作田　康夫
(72)発明者　島田　健太郎
神奈川県川崎市麻生区王禅寺１０９９番地
株式会社日立製作所システム開発研究所内
(72)発明者　橋本　顕義
神奈川県川崎市麻生区王禅寺１０９９番地
株式会社日立製作所システム開発研究所内

Fターム(参考)　5B065 BA01　CA11　CA30　ZA01
5B082 FA04　FA16　HA00

(54)【発明の名称】論理分割可能な記憶装置及び記憶装置システム

(57)【要約】
【課題】
複数の計算機に使用されるNASにおいて、計算機相互の干渉を排除し、入出力の能力の保証・データ破壊や障害の局所化を行い、また装置内部のプロセッサ、メモリ等の資源の利用率を高める。
【解決手段】
ホストインターフェース部100、ファイル入出力要求を受け取ってデータ入出力要求に変換するファイル制御プロセッサ101、変換情報を格納するファイル制御メモリ102、ディスクドライブ群107、ディスク制御プロセッサ104、ディスクドライブ群とディスク制御プロセッサを接続するディスクIF部106、キャッシュメモリ105、プロセッサ間通信接続部103を備え、それぞれを論理的に分割することにより、仮想的な２個以上のNASとして動作させる。
【選択図】　図１

図 1



【特許請求の範囲】
【請求項１】
ネットワークに接続される記憶装置であって、
前記ネットワークと接続され、かつファイルアクセスを受け付けるインターフェースと、
複数のディスクドライブと、
前記ファイルアクセスをブロックアクセスへ変換し、前記ブロックアクセスに基づいて前記複数のディスクドライブを制御する制御部とを有し、
前記制御部が、
前記インターフェース、前記複数のディスクドライブ及び前記制御部を論理的に分割し、複数の仮想的な記憶装置として独立に動作させることを特徴とする記憶装置。
【請求項２】
前記制御部は更にキャッシュメモリを有し、前記キャッシュメモリが前記複数の仮想的な記憶装置の各々に論理的に分割されて割り当てられていることを特徴とする請求項１記載の記憶装置。
【請求項３】
前記制御部は更に前記ファイルアクセスを前記ブロックアクセスへ変換する第一のプロセッサと前記ブロックアクセスに基づいて前記複数のディスクドライブを制御する第二のプロセッサとを有し、
前記第一のプロセッサ及び前記第二のプロセッサが各々論理的に分割され、前記複数の仮想的な記憶装置の各々に割り当てられていることを特徴とする請求項２記載の記憶装置。
【請求項４】
前記第一のプロセッサが、前記インターフェース及び該第一のプロセッサの論理分割を行う第一のハイパバイザを実行し、
前記第二のプロセッサが、前記キャッシュメモリ、複数のディスク装置及び該第二のプロセッサの論理分割を行う第二のハイパバイザを実行することを特徴とする請求項３記載の記憶装置。
【請求項５】
前記制御部はさらに、前記第一のプロセッサが使用するメモリ及び前記第一のプロセッサと前記第二のプロセッサとを接続する通信網を有し、
前記メモリは前記第一のハイパバイザによって論理的に分割され、前記通信網は前記第二のハイパバイザによって論理的に分割されることを特徴とする請求項４記載の記憶装置。
【請求項６】
前記第一のプロセッサ及び前記第二のプロセッサが、前記インターフェース、前記第一のプロセッサ、前記キャッシュメモリ、前記第二のプロセッサ及び前記複数のディスクドライブの論理分割を行うハイパバイザを実行することを特徴とする請求項３記載の記憶装置。
【請求項７】
前記制御部が、前記インターフェース、該制御部及び前記複数のディスクドライブの論理分割を行うハイパバイザを実行することを特徴とする請求項１記載の記憶装置。
【請求項８】
更に管理端末と接続され、
前記制御部は、前記管理端末から入力された情報に基づいて前記論理分割を行うことを特徴とする請求項３記載の記憶装置。
【請求項９】
前記管理端末に入力される情報が、該記憶装置を使用する上位装置がデータ転送速度を重視するという情報であれば、前記複数の仮想的な記憶装置のうち、前記上位装置が使用する仮想的な記憶装置に対する前記キャッシュメモリの割り当てを増加させることを特徴

とする請求項８記載の記憶装置。
【請求項１０】
前記管理端末に入力される情報が、該記憶装置を使用する上位装置が広範囲のランダムなアクセスをするという情報であれば、前記複数の仮想的な記憶装置のうち前記上位装置が使用する仮想的な記憶装置に対する前記キャッシュメモリの割り当てを減少させることを特徴とする請求項８記載の記憶装置。
【請求項１１】
更に管理端末と接続され、
前記制御部は、前記管理端末から入力された情報に基づいて前記論理分割を行うことを特徴とする請求項５記載の記憶装置。
【請求項１２】
前記管理端末に入力される情報が、該記憶装置を使用する上位装置が逐次的な連続したアクセスをするという情報であれば、前記複数の仮想的な記憶装置のうち前記上位装置が使用する仮想的な記憶装置に対する前記キャッシュメモリ及び前記メモリの割り当てを増加させることを特徴とする請求項１１記載の記憶装置。
【請求項１３】
前記管理端末に入力される情報が、該記憶装置を使用する上位装置が少数の大容量のファイルをアクセスをするという情報であれば、前記複数の仮想的な記憶装置のうち前記上位装置が使用する仮想的な記憶装置に対する前記第一のプロセッサの割り当て量を減少させ、前記第二のプロセッサの割当量を増加させることを特徴とする請求項８記載の記憶装置。
【請求項１４】
前記管理端末に入力される情報が、該記憶装置を使用する上位装置が多数の小容量のファイルをアクセスをするという情報であれば、前記複数の仮想的な記憶装置のうち前記上位装置が使用する仮想的な記憶装置に対する前記第一のプロセッサの割り当て量を増加させ、前記第二のプロセッサの割当量を減少させることを特徴とする請求項８記載の記憶装置。
【請求項１５】
前記管理端末に入力される情報が、該記憶装置を使用する上位装置が大容量のファイルを逐次的にアクセスするという情報であれば、前記複数の仮想的な記憶装置のうち前記上位装置が使用する仮想的な記憶装置に対する前記通信網の論理的な割当量を減少させることを特徴とする請求項１１記載の記憶装置。
【請求項１６】
ネットワークと接続され、かつファイルアクセスを受け付けるインターフェースと、複数のディスクドライブと、前記ファイルアクセスをブロックアクセスへ変換し、前記ブロックアクセスに基づいて前記複数のディスクドライブを制御する制御部とを有する記憶装置と、
前記記憶装置と接続される管理端末を有する記憶装置システムであって、
前記記憶装置は、前記管理端末に入力される情報に基づいて、前記インターフェース、前記複数のディスクドライブ及び前記制御部を論理的に分割し、複数の仮想的な記憶装置として独立に動作することを特徴とする記憶装置システム。
【請求項１７】
前記管理端末に入力される情報は、前記記憶装置を使用する計算機のアクセス特性についての情報であり、前記記憶装置は、前記管理端末に入力される前記アクセス特性についての情報に基づいて前記記憶装置が有する資源の論理分割量を計算し、その結果を用いて前記論理的な分割を行うことを特徴とする請求項１６記載の記憶装置システム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ネットワークに接続して用いる記憶装置、特にNASに関する。

【背景技術】
【０００２】
情報処理システムの能力を向上するための方法として、単純に情報処理システムが有する計算機の台数を増やす方法がある。しかし、計算機を多数設置すると個々の計算機の管理に手間がかかり、またこれらの複数の計算機の設置面積や電力等の合計も非常に大きくなるという問題があった。これを解決するために、処理能力の大きい計算機を設置し、その計算機の資源を論理的に複数に分割し、その分割された部分の各々を仮想的な計算機として各々独立に使用する技術が考案されている。これを計算機の論理分割(Logical Partitioning: LPAR)と呼ぶ。例えば特許文献１にこのような論理分割技術の例が開示されている。
【０００３】
論理分割によって一つの計算機を多数の計算機に仮想的に見せることにより、仮想的な計算機の各々に種々のオペレーティング・システムを自由に搭載し、運用・停止や障害処理も仮想的な計算機毎に独立して行えるなど、柔軟な運用が可能となる。また物理的な装置台数が少ないので、装置管理、設置面積、電力等で有利である。但し、従来の計算機におけるLPARでは、計算機内のプロセッサ、メモリなどの資源が論理分割されてそれぞれの仮想的な計算機に割り付けられていたが、計算機に接続される記憶装置については、その記憶装置が有する記憶領域が分割されて仮想的な計算機に各々割り当てられるだけで、それ以上の考慮は特にされていなかった。
【０００４】
一方、記憶装置には、１台の計算機（以下「ホスト」と称することもある）に直結して用いられる形態のほかに、ネットワークを介して複数の計算機に共有される形態がある。この形態において特に、ファイルシステムのインターフェースを有する、すなわち計算機からファイルアクセスが可能な記憶装置をネットワークアタッチドストレージ（以下「ＮＡＳ」）と称する。
【０００５】
ＮＡＳとホストとのデータのやり取りは、ホスト上で動作するオペレーティング・システムが認識する名前や構造を持ったファイルという単位で行われる。このため、ＮＡＳは、データを記憶するためのディスクドライブやその制御部に加え、ホストとのファイル入出力をディスクドライブとのデータ入出力に変換するために使用するプロセッサやメモリを有する。
【０００６】
【特許文献１】特開２００３－１５７１７７号公報
【発明の開示】
【発明が解決しようとする課題】
【０００７】
ＮＡＳは、もともと複数のホストにそれぞれ記憶装置を別個に設置するよりネットワークで共有する記憶装置を設けると有利であるという思想に基づいている。このため、複数のホスト間で記憶領域等を共有するための制御をＮＡＳ自身が行わなければならなかった。またあるホストが大量にデータの読み書きを行うと、ＮＡＳの処理能力の大半が消費され、他のホストに対するデータの入出力の能力が低下する。更に、あるホストの誤操作等でＮＡＳのデータ破壊や障害が発生すると、他のホストが使用するデータにも影響を及ぼす場合もあった。
【０００８】
本発明が解決しようとする課題は、ＮＡＳが複数のホストから共有されるときに、共有にかかわる制御を削減するとともに、相互の計算機の干渉を排除し、入出力の能力の保証・データ破壊や障害の局所化を行うことである。また本発明が解決しようとする別の課題は、ＮＡＳ内のプロセッサ、メモリ等の資源の利用率を高めることである。
【課題を解決するための手段】
【０００９】

上記の課題を解決するために、本発明では、ＮＡＳにおいて論理分割を行う。　より具体的には、ネットワークに接続され、ファイルアクセスを受け付ける記憶装置において、記憶装置が有する各資源、例えばディスクドライブ、ネットワークとのインターフェース、ファイルアクセスを制御するプロセッサ等を、記憶装置が有する制御部が論理的に分割して、個々の論理区画（仮想的な記憶装置）を独立して動作させる構成とする。

【００１０】

尚、制御部は複数のプロセッサを有し、これらのプロセッサが論理分割を分担して行ったり、全体として行ったりしても良い。

更に、管理端末を有し、論理分割に必要な情報をこの管理端末から入力する構成としても良い。この場合、管理端末には記憶装置を使用する計算機のアクセス特性のみが入力され、管理端末がそのアクセス特性から論理分割に必要な情報を算出して記憶装置に伝達する構成としても良い。

更に、記憶装置を使用する計算機が管理端末を兼ねる構成でも良い。

【発明の効果】

【００１１】

本発明による記憶装置では、複数のホストから共有されるときに、共有にかかわる制御を削減するとともに、ホスト相互の干渉を排除し、データの入出力の能力の保証・データ破壊や障害の局所化を行うことができる。

又、ホスト毎またはホストのグループ毎に独立したユーザ認証を行うことができる。また本発明により、記憶装置内のプロセッサ、メモリ、記憶媒体等の資源の利用率を高めることが可能である。

【００１２】

更に、１台のＮＡＳで複数の仮想的なNASを提供することができ、ホストのオペレーティング・システムの自由度が増し、運用・停止や障害処理も独立して行うことが可能であり、装置管理、設置面積、電力等で有利となる。

【発明を実施するための最良の形態】

【００１３】

以下、本発明の実施の形態を図面を用いて説明する。尚、本発明が下記の実施形態の記載に限定されることが無いのは言うまでもない。

図１は、本発明が適用されたＮＡＳの実施形態一例を示す図である。ＮＡＳは、ホスト（以下「上位装置」とも称する）と接続される二つのホストインターフェース部100、ホストからのファイル単位の入出力要求をブロック単位のデータの入出力要求に変換する三つのファイル制御プロセッサ101、ファイル単位の入出力要求をブロック単位のデータの入出力要求に変換するために必要な情報（以下「変換情報」とも称する）を格納する二つのファイル制御メモリ102、記憶媒体である四つのディスクドライブ群107、ディスクドライブ群107とのデータの入出力の制御を行う三つのディスク制御プロセッサ104、ディスクドライブ群107に入出力されるデータを一時的に蓄える二つのキャッシュメモリ106、ディスク制御プロセッサ104とディスクドライブ群107とを接続する二つのディスクIF部106、ディスク制御プロセッサ104とファイル制御プロセッサ101とを接続する二つのプロセッサ間通信接続部103を有する。

【００１４】

ここで、ディスクドライブ群107には、複数のディスクドライブが含まれ、各グループ毎にRAID構成がとられている場合もある。また、「ブロック」とは、ディスク制御プロセッサ104がディスクドライブへデータを格納する所定の単位であり、一般的に512Bが採用される。又、「変換情報」とは、ファイルシステムで使用されるファイル名及びファイルの先頭からの位置とブロックとの対応関係を示す情報であり、一般的にはI-Node等のリンク構造や、アドレス変換テーブルのようなデータ構造で表現される。

【００１５】

尚、本明細書においては、同一の装置には同一の番号を付し、同一の装置を区別するときは英字のａ、ｂ等を番号に付与する。又、上述した各装置の個数も例示であり、本発明

を限定するものではない。

【００１６】

図１において、上位装置からＮＡＳに送信されたファイル単位の入出力要求は以下のようにＮＡＳで処理される。

ＮＡＳにはまず初めに、ホストからファイル名を指定したファイル使用開始（オープン）処理の要求が送信される。次に実際のデータの入出力要求が送信され、最後にファイル使用終了（クローズ）処理の要求が送信される。

【００１７】

これらの要求は、いずれかのホストインターフェース部100で受信され、いずれかのファイル制御プロセッサ101に渡される。ファイル制御プロセッサ101は、ファイル制御メモリ102に格納された変換情報を参照して、ホストに要求されたファイル名を確認し、使用開始されたファイル名を記録し、該ファイルに対するデータ入出力要求を、データが格納されているディスクドライブ群107へのデータ入出力要求に変換する。

【００１８】

変換されたデータ入出力要求は、いずれかのプロセッサ間通信接続部103を介して、いずれかのディスク制御プロセッサ104に送出される。またファイル制御メモリ102に必要な変換情報が格納されていない場合は、ファイル制御プロセッサ101は、プロセッサ間通信接続部103を介して、いずれかのディスク制御プロセッサ104にディスクドライブ群107の予め定められた記憶領域に格納されている変換情報を要求する。

【００１９】

ディスク制御プロセッサ104は、いずれかのファイル制御プロセッサ101よりプロセッサ間通信制御部103を介して受け取ったデータ入出力要求（変換情報の要求も含む）に対し、当該データがいずれかのキャッシュメモリ105に格納されていないかどうか調べる。当該データがいずれかのキャッシュメモリ105に格納されていた時には、ディスク制御プロセッサ104は、要求されたデータの書き込み又は読み出しをキャッシュメモリ105に対して行う。

【００２０】

その後、ディスク制御プロセッサ104は、書込みの時には完了したという結果を、読み出しの時には読み出しの完了と読み出したデータとを合わせて、プロセッサ間通信部103を介して、入出力要求を送信したファイル制御プロセッサ101に返送する。ファイル制御プロセッサ101は、返送された結果及びデータを処理し、入出力要求を受信したホストインターフェース部100を介して、ＮＡＳに入出力要求を送信した上位装置に結果（データあるいは処理完了の報告等）を送信する。

【００２１】

一方、ファイル制御プロセッサ101から要求されたデータが全てのキャッシュメモリ105に格納されていなかった場合、ディスク制御プロセッサ104は、要求されたデータがディスクドライブ群107のどの部位に格納されているかを特定し、いずれかのディスクIF部106を介して、ディスクドライブ群107よりデータを読み出して、いずれかのキャッシュメモリ105に格納する。

【００２２】

その後、ディスク制御プロセッサ104は、要求されたデータの読み出し又は書き込みをデータが格納されたキャッシュメモリ105に対して行う。それ以降の処理は、上述と同様である。

【００２３】

尚、キャッシュメモリ105に格納されているデータは、一定時間経過又はキャッシュメモリ105の空き領域が不足した時などにディスクドライブ群107に書き戻される。

【００２４】

本実施形態において、上述した処理は、例えば図１に示したような論理分割境界によって分割された単位（論理区画）によって、それぞれ独立して行われる。それぞれの論理区画に割り当てられる処理のための物理的な資源であるホストインターフェース部100、フ

ファイル制御プロセッサ101、ファイル制御メモリ102、プロセッサ間通信接続部103、ディスク制御プロセッサ104、キャッシュメモリ105、ディスクIF部106及びディスクドライブ群107は、各論理区画に一度割り当てられるとその論理区画の処理に専ら用いられる。具体的に言えば、図１において違う論理区画に割り当てられたファイル制御プロセッサ101aとディスク制御プロセッサ104cとは、上述したデータの入出力要求の遣り取りを行わない。

【００２５】

また図１で論理分割境界をまたがっている表示されている資源（例えばファイル制御メモリ102b）は、その容量等が予め割り当てられた比率で論理的に分割されて各論理区画毎に用いられる。このようにすることにより、各論理区画がそれぞれ独立した仮想的なNASとして動作する。

【００２６】

各論理区画への物理資源の論理的な分割・割り当て処理は、実際にはファイル制御プロセッサ101やディスク制御プロセッサ104で実行される。論理分割の制御の方法としては、以下の二つの方法が考えられる。

【００２７】

一つ目は、ファイル制御プロセッサ101とディスク制御プロセッサ104が幾つかの物理資源の論理分割の制御を分担して行い、全体としてはお互いが連携して論理分割を制御する方法である。

例えば、ホストインターフェース部100、ファイル制御プロセッサ101及びファイル制御メモリ102の割り当て処理を、ファイル制御プロセッサ101が行う。この処理を以下「ファイル制御ハイパバイザ」と呼ぶ。

【００２８】

またプロセッサ間通信接続部103、ディスク制御プロセッサ104、キャッシュメモリ105、ディスクIF部106及びディスクドライブ群107の割り当て処理を、ディスク制御プロセッサ104が行う。この処理を以下「ディスク制御ハイパバイザ」と呼ぶ。ファイル制御プロセッサ101で実行されるファイル制御ハイパバイザと、ディスク制御プロセッサ104で実行されるディスク制御ハイパバイザは、互いに連携して、それぞれの割り当て処理を行う。連携の具体的な内容については、後述する。尚、ファイル制御ハイパバイザを実行するファイル制御プロセッサ101はいずれか一つ、例えばファイル制御プロセッサ101aのみであってもよく、又は複数、例えばファイル制御プロセッサ101aおよびｂであっても良い。これはディスク制御ハイパバイザでも同様である。

【００２９】

二つ目は、二つの制御プロセッサが共同して全体の物理資源の論理分割を制御する方法である。具体的には、ＮＡＳの全ての資源の論理区画の割り当て処理（以下「統合ハイパバイザ」）を、ファイル制御プロセッサ101a～101c、ディスク制御プロセッサ104a～104cのすべてで行う。

【００３０】

具体的には、各プロセッサ上で動作するハイパバイザは、例えば次のようにして論理分割を実現する。

まず、各プロセッサ上で動作する基本ＩＯ処理ソフトウェア（ＢＩＯＳ）に対し、当該プロセッサが割り当てられた論理区画内のＩＯ処理資源以外の資源が見えないようにする。例えば、図１でファイル制御プロセッサ101aは物理的にはホストインターフェース部100bと接続されているが、図示されている点線で論理区画が分かれる設定がされた場合は、ホストインターフェース部100bを見えないようにする。

【００３１】

より具体的には、ＢＩＯＳ内で当該プロセッサに接続されている資源や利用可能な資源を調べるための特権命令が実行された場合、当該特権命令の実行でソフトウェア的な割り込みを発生させ、ハイパバイザに実行を移す。ハイパバイザ内で当該プロセッサが属している論理区画に対して割り当てられている資源を調べ、その論理区画に割り当てられてい

る資源のみが接続されているように当該特権命令の結果をセットし、割り込みを発生したＢＩＯＳへ復帰する。

【００３２】

このようにすることにより、各プロセッサはそのプロセッサの属する論理区画の資源のみを扱うことになり、論理区画間の分離が実現される。

資源の中にはメモリや複数の通信チャネルを備えるホストインターフェース部100やプロセッサ間通信部103があるが、このような場合は、それぞれの論理区画のプロセッサに見せる資源の量（メモリなら開始物理アドレス、終了物理アドレスでメモリの容量、通信チャネルならチャネルの物理番号の組で示されるチャネル数）を制御すればよい。

【００３３】

またプロセッサ自身については、それぞれのプロセッサが一つの論理区画に完全に割り当てられている場合には、当該プロセッサはその論理区画の処理で占有させれば良い。

一方、ある一つのプロセッサを二つ以上の論理区画に割り当てて各々の共有割合を定めて共有させることも考えられる。このような場合は、タイマー割り込みをハードウェアで各プロセッサに実装し、そのタイマー割り込みでハイパバイザが一定時間ごとに起動されるようにする構成が考えられる。

【００３４】

上述のタイマー割り込みで起動されたハイパバイザは、当該プロセッサでそれぞれの論理区画の処理をどのくらい行ったかを計測し、予め定めた共有割合に従って次に処理を行うべき論理区画を決定して、その論理区画の処理へプロセッサの実行を移す。このようにすれば、一つのプロセッサを時間的に予め定めた割合に分割して、二つ以上の論理区画に割り当てることができる。

【００３５】

なお、ハイパバイザの実現方法としては、上記の例のほかにも、例えば、各プロセッサに接続され、資源管理を行う専用のハードウェアや、小規模のマイクロプログラムで制御されるような専用のコプロセッサ等を搭載して、論理分割の制御を実現してもよい。

【００３６】

又、論理分割に関する情報、例えば論理区画１で使用されるプロセッサ、メモリ、通信接続部等を指定する情報等は、ファイル制御メモリ102、キャッシュメモリ105、ディスク群107のディスクドライブ又はその他の記憶媒体のいずれか一つ又は複数に格納されており、各ハイパバイザは、その情報を読み出すことによって、論理分割の指定をBIOS等に対して行う。尚、この情報は、後述する管理端末等を介して設定される。

【００３７】

上記のようにして実現されたハイパバイザについて、先に述べたように、ファイル制御プロセッサでファイル制御ハイパバイザ、ディスク制御プロセッサでディスク制御ハイパバイザを動作させる場合は、ファイル制御ハイパバイザは、ホストインターフェース部100、ファイル制御プロセッサ101及びファイル制御メモリ102の割り当て処理を、ディスク制御ハイパバイザは、プロセッサ間通信接続部103、ディスク制御プロセッサ104、キャッシュメモリ105、ディスクIF部106及びディスクドライブ群107の割り当て処理を行い、二種類のハイパバイザを連携させるようにする。

【００３８】

具体的には、後述するような管理端末等によって論理分割の指定を行う際に、ファイル制御ハイパバイザに対する指定とディスク制御ハイパバイザに対する指定を関連付けて行われるようにする。あるいは、論理分割に与える指定の仕方によっては、後述するように、ファイル制御ハイパバイザとディスク制御ハイパバイザが、当該論理区画が指定された論理分割に関する要求に合致するように、互いに自動的に調整し合うようにする。

【００３９】

統合ハイパバイザの場合には、例えば、統合ハイパバイザを起動する各プロセッサが、すべての資源の各論理区画への割り当て情報を共有し、当該ハイパバイザが、起動したプロセッサが用いる資源をその割り当て情報参照して決定し、割り当て処理を実行する。

【００４０】
次に、ＮＡＳの各資源の論理区画への割り当ての具体例を説明する。以下、図１で示したＮＡＳを二つの論理区画（論理区画１及び論理区画２）に論理分割した場合を例として説明する。しかし、論理区画の数は幾つでも構わない。又、以下ではファイル制御ハイパバイザとディスク制御ハイパバイザが連携して論理分割を行うとして説明するが、統合制御ハイパバイザでも構わない。更に、ハイパバイザを主語とした場合は、その処理は実際は各ハイパバイザの処理を行うプロセッサによって実行されているものとする。
【００４１】
図２は、ファイル制御メモリ102及びキャッシュメモリ105の論理区画への割り当ての例を示す図である。例えば論理区画１を使用する上位装置の要求がデータ読み出し速度重視であるときは、ディスク制御ハイパバイザは、論理区画１へのキャッシュメモリ105の割り当て量を増やし、要求されたデータができるだけキャッシュメモリ105に格納されるようにする。このときはファイル制御メモリ102の論理区画１への割り当て量は少なくてもよい。
【００４２】
これに対応して、ファイル制御ハイパバイザは、論理区画１へのファイル制御メモリ102の割り当て量を下げ、論理区画２により多く記憶容量を割り当てる。これにより、NAS全体でのファイル制御メモリ102の利用率を向上させることが可能である。
【００４３】
一方、論理区画１を使用する上位装置の要求が応答速度重視であれば、変換情報がなるべくファイル制御メモリ102に格納されるように、ファイル制御ハイパバイザは、論理区画１により多くのファイル制御メモリ102の記憶容量を割り当てる割り当て処理を行う。このときは、論理区画１へのキャッシュメモリ105の割り当て量は少なくてもよい。この場合、ディスク制御ハイパバイザは論理区画２により多くのキャッシュメモリ105の記憶容量を割り当てることができ、ＮＡＳ全体として、キャッシュメモリ105の利用率を向上することが可能である。
【００４４】
又、論理区画１を利用する上位装置の入出力要求が、ＮＡＳが有するディスクドライブ群107の広範囲に散らばるデータへのランダムなアクセスが主体の場合、論理区画１にファイル制御メモリ102及びキャッシュメモリ105の多くの記憶容量を割り当てても広範囲に散らばるアクセスのすべての情報を格納することは難しいので、その割り当ての効果が薄い。従って、このような場合には、ファイル制御ハイパバイザとディスク制御ハイパバイザがファイル制御メモリ102及びキャッシュメモリ105の論理区画１への割り当て量を少なくし、他の論理区画である論理区画２に記憶容量を多く割り当て、キャッシュメモリ105等の利用率を向上させる。
【００４５】
逆に論理区画１を利用する上位装置の入出力要求が、ＮＡＳが有するディスクドライブ群107の連続範囲に格納されたデータへのシーケンシャルなアクセスが主体の場合、ＮＡＳ自体でアクセスに必要な情報及び後に読み出されるデータを前もって特定することが可能である。そのため、それらの情報やデータを充分ファイル制御メモリ102及びキャッシュメモリ105に格納することができるように、ファイル制御ハイパバイザ及びディスク制御ハイパバイザが、ファイル制御メモリ102及びキャッシュメモリ105の論理区画１への割り当て量を増やすように割り当て処理を行うことが考えられる。
【００４６】
図３は、ファイル制御プロセッサ101及びディスク制御プロセッサ104の論理区画への割り当ての例を示す図である。
論理区画１を使用する上位装置からの入出力要求が少数の大容量ファイルアクセスの場合、ファイル制御プロセッサ101で実行されるファイル入出力をデータ入出力に変換する処理の量はあまり多くない。従って、ファイル制御プロセッサ101の論理区画１への割り当て量は少なくてもよい。

【００４７】
この場合、ファイル制御ハイパバイザが論理区画１へのファイル制御プロセッサ101の論理区画１への割り当て量（具体的にはプロセッサの占有率）を下げ、他の論理区画である論理区画２に論理区画１に比較して相対的に多くプロセッサ資源を割り当てることにより、ＮＡＳが有するファイル制御プロセッサ101の利用率の向上が可能になる。
【００４８】
又この場合は、ファイルのデータ量が大きいため、論理区画１に割り当てられたディスク制御プロセッサ104で実行されるデータ入出力の処理の量は多くなる。したがって、ディスク制御ハイパバイザは、論理区画１へのディスク制御プロセッサ104の割り当て量を増やすように割り当て処理を行う。
【００４９】
また、論理区画１を使用する上位装置の入出力要求が多数の小容量ファイルアクセスである場合、ファイル制御プロセッサ101で実行されるファイル入出力をデータ入出力に変換する処理の量は多くなる。そこで、ファイル制御ハイパバイザは、論理区画１へのファイル制御プロセッサ101の割り当て量を増やすように割り当て処理を行う。
【００５０】
このとき、ファイルのデータ量自体は小さいので、論理区画１に割り当てられたディスク制御プロセッサ104で実行されるデータ入出力の処理の量はあまり多くない。そこで、ディスク制御ハイパバイザは、論理区画１へのディスク制御プロセッサ104の割り当て量を減らして、論理区画２へのディスク制御プロセッサ104の割り当て量を増やす。これにより、ＮＡＳにおけるディスク制御プロセッサ104の利用率の向上が可能となる。
【００５１】
さらに、論理区画１を使用する上位装置が高性能のＮＡＳを必要としない場合は、ファイル制御ハイパバイザ及びディスク制御ハイパバイザは、ファイル制御プロセッサ101及びディスク制御プロセッサ104の論理区画１への割り当て量を減らすように割り当て処理を行う。逆に、論理区画１を使用する上位装置が高性能のＮＡＳを必要とする場合は、ファイル制御ハイパバイザ及びディスク制御ハイパバイザは、ファイル制御プロセッサ101及びディスク制御プロセッサ104の論理区画１への割り当て量を増やすように割り当て処理を行う。
【００５２】
図４は、プロセッサ間通信接続部103の論理区画への割り当ての例を示す図である。論理区画１を使用する上位装置からの入出力要求が大容量のシーケンシャルアクセスである場合、ディスク制御ハイパバイザは、論理区画１へのプロセッサ間通信接続部103の割り当て量（具体的には通信帯域）を増やし、ファイル制御プロセッサ101とディスク制御プロセッサ104との間のデータ通信能力（言い換えると、ホストインターフェース部100からキャッシュメモリ105までの間のデータ通信能力）を確保するように割り当て処理を行う。
【００５３】
また、論理区画１を使用する上位装置の入出力要求が小容量のシーケンシャルアクセスであれば、論理区画１へのプロセッサ間通信接続部103の割り当て量は大きくなくてもよい。さらに上位装置からの要求がランダムアクセスであれば、上位装置からみた論理区画１で構成される仮想的なＮＡＳの性能はプロセッサ間通信接続部103の論理区画１への割り当て量にあまり影響されない。従ってこれらの場合は、ディスク制御ハイパバイザは、論理区画１へのプロセッサ間通信接続部103の割り当て量を減らして他の論理区画（ここでは論理区画２）へ割り当てを増やし、ＮＡＳにおけるプロセッサ間通信接続部103の利用率を高めるように割り当て処理を行う。
【００５４】
図５は、ディスクドライブ群107の論理区画への割り当ての例を示す図である。論理区画１を使用する上位装置が記憶容量優先であれば、ディスク制御ハイパバイザは、記憶容量効率の高いRAID5構成（図７ではデータが格納されているディスクドライブ３個に対し

パリティが格納されているディスクドライブ１個の割合で容量効率は75%となる）になっているディスクドライブ群701を論理区画１に割り当てるように処理を行う。このとき、ディスクドライブの回転速度も例えば毎分7、500回転など、あまり速いものでなくてもよい。

【００５５】

一方、論理区画１を使用する上位装置がアクセス性能を重視する場合であれば、ディスク制御ハイパバイザは、アクセス性能を高められるRAID1構成（図７において、同一のデータが複製されて２つのディスクドライブに格納されるので、記憶容量効率は50%だが、同一のデータに対して２個のディスクドライブのいずれも使用可能なので、トータルのアクセス性能は一つのディスクドライブの２倍になる）になっているディスクドライブ群701を論理区画１に割り当てるように処理を行う。尚、この場合、ディスク制御ハイパバイザは、ディスクドライブ群107に含まれるディスクドライブの回転速度も考慮して、同じRAID１構成のディスクドライブ群107のうち、高回転速度、例えば毎分15、000回転のディスクドライブを有するディスクドライブ群107を論理区画１に割り当てる処理を行っても良い。

【００５６】

尚、ホストインターフェース部100の論理区画への割り当ては、各論理区画を使用する上位装置から論理区画に要求される性能に応じて、ファイル制御ハイパバイザが割り当てる。具体的には、上位装置の要求性能が高い場合には、ファイル制御ハイパバイザは、上位装置が使用する論理区画に大きな割り当て量、即ち上位装置との間の高い通信能力（通信帯域等）を割り当てる。一方、上位装置の要求性能が低いあるいは特に要求が無い場合には、ファイル制御ハイパバイザは、上位装置が使用する論理区画に小さな割り当て量、即ち上位装置との間の低い通信能力（通信帯域等）を割り当て、NAS全体の効率を重視することが考えられる。

【００５７】

更に本実施形態のように１つのNASを論理的に分割して用いることにより、NASにおける上位装置のユーザ認証を、論理区画単位毎に独立して行うことが可能である。図６はその概念の例を示す図である。

本図では、論理区画１を使用する上位装置Aを識別子（以下「ID」）＝“abc”であるユーザAとID=“def”であるユーザBが使用し、論理区画２を使用する上位装置BをID=“ghi”であるユーザCとID=“abc”であるユーザDが使用する。このとき上位装置Aのユーザと上位装置Bのユーザが同じID=“abc”であるので、このユーザAとユーザDをNASで区別して扱うためには、従来は上位装置または上位装置のグループにIDを付与してそのIDとユーザのIDを組み合わせてユーザを区別するなどの特別な処理が必要であった。

【００５８】

しかし本実施形態では、ホストインターフェース部100及びファイル制御プロセッサ101が論理的に分割され、各論理区画が別々の仮想的なNASのホストインターフェース部100及びファイル制御プロセッサ101として動作するので、ユーザ認証も各論理区画毎に独立して行われる。即ち、図８において同一のID=“abc”を持つユーザAとユーザDは、別々の論理区画でユーザ認証されるので、自然に区別して扱われ、ユーザAとユーザDを区別するために特別な処理をなんら必要としない。つまり、論理区画さえ異なれば、特別な処理をすることなく、複数のユーザに同一のIDを与えることができる。

【００５９】

さらには、各論理区画に割り当てられているホストインターフェース部100及びファイル制御プロセッサ101の資源は他の論理区画で使用されることは無いので、ある論理区画のユーザが大量のデータアクセスを行っても、ほかの論理区画のユーザは影響を受けることがない。

【００６０】

次に、第二の実施形態について説明する。本実施形態のNASは、上述したNASの、ファイル制御プロセッサ101及びディスク制御プロセッサ104並びにファイル制御メモリ102及び

キャッシュメモリ105を統合した、それぞれ一種類のプロセッサ、メモリを有する。

図７は、第二の実施形態の構成の例を示す図である。図９において、統合制御プロセッサ901は、ファイル制御プロセッサ101及びディスク制御プロセッサ104を統合したプロセッサであり、統合メモリ902は、ファイル制御メモリ102及びキャッシュメモリ105を統合したメモリである。

【００６１】

上述した実施形態（図１）に比べ、本実施形態では、プロセッサ間通信接続部103が不要となり、装置構成が簡単になる。図７において、上位装置からのファイル単位の入出力要求をデータの入出力要求に変換する処理と、ディスクIF部106とディスクドライブ群107との間のデータの入出力の制御は、両方とも統合プロセッサ901で行われる。また、統合メモリ902には、変換情報及びディスクドライブ群107のデータが格納される。図７における他の部分の構成・動作は図1と同じである。

【００６２】

図７においても、図１と同様に、例えば図７中に示したような論理分割境界によって分割された論理区画で、それぞれ独立して処理が行われる。それぞれの論理区画に割り当てられる処理のための物理的な資源であるホストインターフェース部100、ホストインターフェース部100、統合制御プロセッサ901、統合メモリ902、ディスクIF部106及びディスクドライブ群107は、各論理区画に一度割り当てられるとその論理区画の処理に専ら用いられる。このようにすることにより、各論理区画がそれぞれ独立した仮想的なNASとして動作する。

【００６３】

本実施形態における各論理区画への物理資源の論理的な分割・割り当て処理は、実際には統合制御プロセッサ901が実行する。統合制御プロセッサ901は、上述した統合制御ハイパバイザの制御を行う。

【００６４】

図８は、NASの論理分割の設定を入力するための管理端末の入力画面の例を示す図である。このような入力画面は、第一の実施形態及び第二の実施形態の双方で使用される。管理者等が図８のような論理分割の設定入力を行ってその内容がNASに通知されることによって、NASで動作するハイパバイザがNASの各資源を論理分割する。より詳細に言えば、設定入力の内容が、NASが有するある記憶領域に格納され、ハイパバイザは、この格納された情報にしたがって論理分割を行う。

【００６５】

このような管理端末は、具体的にはネットワークを通じてNASに接続する上位装置が有していてもよい。また、NASに専用線で接続されるコンソール装置で実現されていてもよい。このようなコンソール装置はキーボード等の入力装置とディスプレイ等の表示装置で実現することができる。

【００６６】

管理者等が入力した情報は、上位装置やコンソール装置から専用のプロトコル又は汎用のプロトコルを用いてNASに転送される。NASはその情報を受取るためのインターフェース（例えばホストインターフェース部100又は専用のインターフェース）を有する。

【００６７】

以下、図８で示される設定入力画面について詳細に説明する。該画面には論理分割数を入力するフィールドがあり、管理者等は、初めにNASの論理分割数をいくつにするかをこのフィールドに入力する（図８の例では３）。この際、管理者等が論理分割数を入力すると、その数に応じた論理区画が物理資源毎に画面に表示され、各論理区画への資源割り当ての初期値が表示される。

【００６８】

その後、管理者等は、各プロセッサ、メモリ等の資源の分配を画面を見ながら入力していく。このとき、例えばファイル制御プロセッサ101とディスク制御プロセッサ104の割り当てを指定する部分では、図示するように、それぞれを各論理区画にどのように割り当て

ているかを関連して設定できるような表示（図ではプロセッサ同士を並べて表示し、関連性がわかりやすいようにしている）にすれば、先に図５で説明したような割り当て制御の設定を入力することが容易になる。

【００６９】

また例えば図示したように、ファイル制御プロセッサ101やディスク制御プロセッサ104の論理区画に対するそれぞれの割り当て量を個別に設定するつまみ（ポインティングデバイスで選択できる部分）のほかに、連動して設定することができるつまみも用意しておく。同様にファイル制御メモリ102とキャッシュメモリ105についても、各論理区画への割り当て量を関連づけて表示し、個別設定・連動設定のつまみを用意しておく。

【００７０】

図８の例では、管理者等は、プロセッサ間通信接続部103について、ファイル制御プロセッサ101とディスク制御プロセッサ104（ホストインターフェース部100とキャッシュメモリ105）の間の全体のデータ転送能力を、それぞれ各論理区画にどの割合で割り当てるかを入力する。

【００７１】

又、管理者等は、ホストインターフェース部100について、資源の割合ではなく物理的に上位装置を接続するためのネットワークの接続口（ポート）を特定することで割り当ての情報を入力する。ただし、単に資源の割合で入力する入力方法でも良い。更に管理者等は、ディスクドライブ群107について、各論理区画に割り当てる物理的なディスクドライブの容量・RAID構成・能力（回転数）を設定することで資源の割り当てを行う。

【００７２】

以上の割り当て設定の入力は例であり、これ以外にも、個別に数値入力することやある程度の自動設定をしてもよい。例えば、上位装置がある論理区画に要求するアクセスの特性（ランダムかシーケンシャルか、１転送あたりの平均的なデータ長、最小データ転送速度、最大レスポンス時間等）を管理者等が管理端末を介して入力することにより、予めいくつか作成してある設定値のセット、具体的には図２～５で示された特徴を有する設定値のセットから入力された特性に合うものをそれぞれハイパバイザで選んで設定するようにしてもよい。

【００７３】

これにより、例えば、管理者がシーケンシャルを指定した場合には、予め作成してある設定値のセットから，それに対応する設定値（図２と図４で示したシーケンシャルに対応した値）をハイパバイザで選んで設定し，論理分割の処理を行う。

【００７４】

具体的には、大容量アクセス向きの論理区画、小容量ファイルアクセス向きの論理区画、中間的な容量のファイルアクセスの論理区画の三つの論理区画を作成したいときには、図８のファイル制御プロセッサ、ディスク制御プロセッサの割り当て設定例で示したような、ファイル制御プロセッサの割り当てがディスク制御プロセッサをより少ない論理区画１、ファイル制御プロセッサの割り当てがディスク制御プロセッサの割り当てより多い論理区画２、同等の割り当て量の論理区画３の割り当て設定値をハイパバイザの参照する領域内（図９に後述する各論理区画への各資源の割り当て情報等を格納しておく記憶領域内）に準備しておく。

【００７５】

管理者等は、割り当て設定を実際に行うに当っては、大容量ファイルアクセス向き、少容量アクセス向き、両者の中間の三つの論理区画といった、論理区画に要求する特性を簡便に指定する。するとハイパバイザがその指定に対応した割り当て設定値を自動的に選んで設定する。

これにより、管理者が所望の性能・特性を有する論理区画を簡単に指定することができる。

【００７６】

また管理者等に割り当て設定の入力は、それぞれの論理区画が正しく動作できるだけの

資源をかならず割り当てるような注意が必要となる。例えば、ファイル制御プロセッサやディスク制御プロセッサの割り当てを０にすることはできない。この点、上述のような自動設定においては、予め割り当て資源量に下限を設け、自動的にこれを守るようにしてもよい。また、図８で図示しているような入力例においては、当該ＮＡＳ装置においてそれぞれの資源の割り当て量の下限を予め定義しておき、これを下回る割り当て量の場合には警告を出したり、そのような割り当て入力をチェックして受け付けないようにしてもよい。

これにより、管理者等は安全に論理区画を設定することが出来る。

【００７７】

図９は、上記のように管理者等が設定した論理区画への各資源の割り当てを示す情報を図示した例である。図９に示すような物理的資源と論理区画との対応関係は、ハイパバイザが管理端末から受信した情報に基づいて作成する。具体的には、それぞれのハイパバイザが、ＮＡＳ装置の物理的資源の構成に関する情報を有しており、管理者等の入力情報とその構成に関する情報に基づいて物理的資源を各論理区画に割り当て、図９のような対応関係を作成する。尚、図９において、縦軸の項目はＮＡＳ装置が有する装置構成によってその項目数が増減し、横軸の論理区画は、管理者等の指定によりその数が変更される。

【００７８】

そして、図９のような対応関係の情報は、先に説明したように、ファイル制御メモリ102、キャッシュメモリ105、ディスク群107のディスクドライブ又はその他の記憶媒体のいずれか一つ又は複数のハイパバイザが専ら用いる領域に格納される。それぞれのハイパバイザは格納された情報を参照して、各論理区画に利用させる資源を決定し、割り当ての処理を行う。

【図面の簡単な説明】

【００７９】

【図１】NASの構成例を示す図である。

【図２】ファイル制御メモリとキャッシュメモリの割り当ての例を示す図である。

【図３】ファイル制御プロセッサとディスク制御プロセッサの割り当ての例を示す図である。

【図４】プロセッサ間通信接続部の割り当ての例を示す図である。

【図５】ディスクドライブ群の割り当ての例を示す図である。

【図６】ホストインターフェース部、ファイル制御プロセッサの論理分割による上位装置のユーザ認証の概念を示す図である。

【図７】NASの構成の例を示す図である。

【図８】NASの論理分割の設定入力画面の例を示す図である。

【図９】NASの各資源の論理分割情報の例を示す図である。

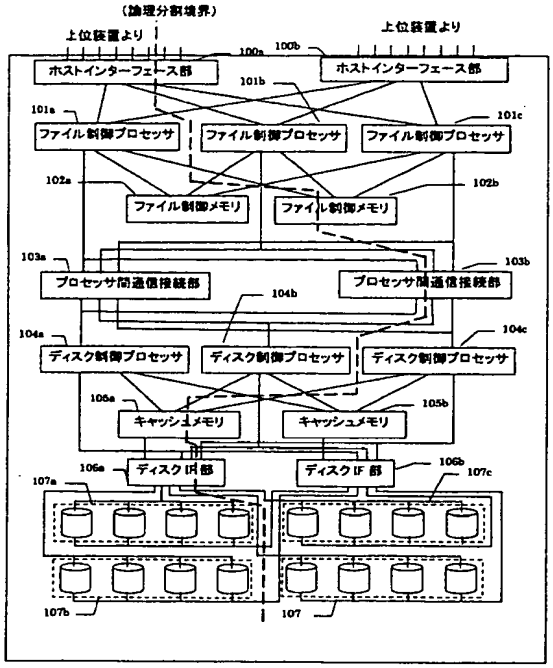
【符号の説明】

【００８０】

100…ホストインターフェース部、101…ファイル制御プロセッサ、102…ファイル制御メモリ、103…プロセッサ間通信接続部、104…ディスク制御プロセッサ、105…キャッシュメモリ、106…ディスクIF部、107…ディスクドライブ群、901…統合制御プロセッサ、902…統合メモリ。

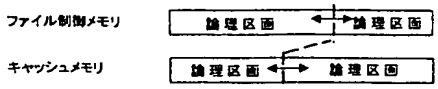
【図１】

図 1



【図２】

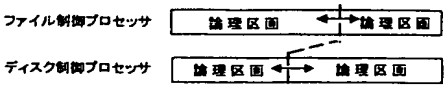
図 2



	データ読み出し速度重視	応答速度重視	広範囲ランダムアクセス	シーケンシャルアクセス
ファイル制御メモリ	割り当て小	割り当て大	割り当て小	割り当て大
キャッシュメモリ	割り当て大	割り当て小	割り当て小	割り当て大

【図３】

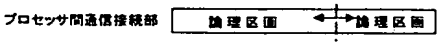
図 3



	少数の大容量ファイルアクセス	多数の小容量ファイルアクセス	低要求性能	高要求性能
ファイル制御プロセッサ	割り当て小	割り当て大	割り当て小	割り当て大
ディスク制御プロセッサ	割り当て大	割り当て小	割り当て小	割り当て大

【図４】

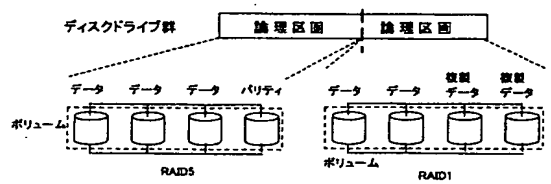
図 4



	大容量シーケンシャルアクセス	少容量シーケンシャルアクセス	ランダムアクセス
プロセッサ間通信接続部	割り当て大	割り当て中	割り当て小

【図５】

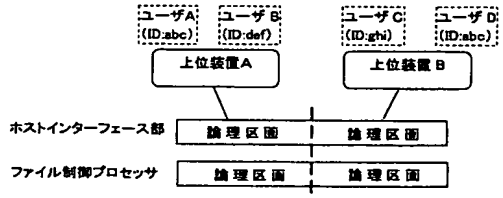
図 5



	容量優先	アクセス性能優先
ディスクドライブ群	RAID5ボリューム割り当て	RAID1ボリューム割り当て

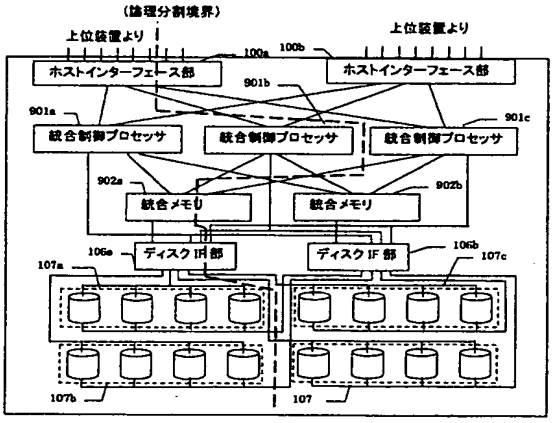
【図６】

図 6



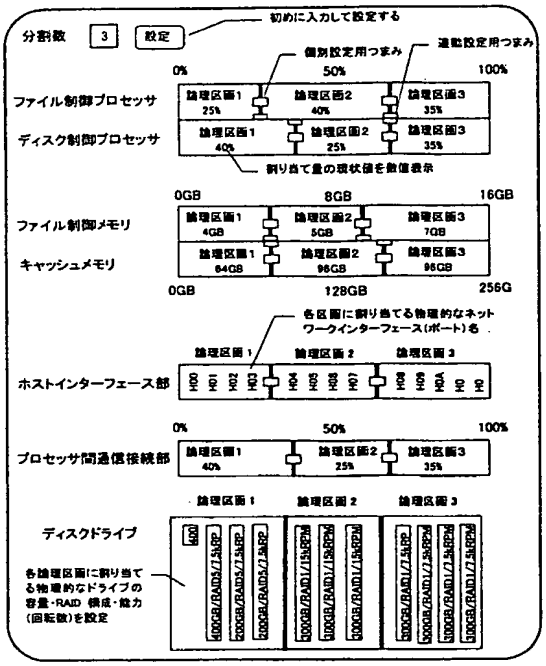
【図７】

図 7



【図８】

図 8



【図９】

図9

	論理区画1	論理区画2	論理区画3	未割り当て
ファイル制御プロセッサ1	100%	0%	0%	0%
ファイル制御プロセッサ2	0%	50%	50%	0%
ファイル制御プロセッサ3	0%	0%	0%	100%
ディスク制御プロセッサ1	100%	0%	0%	0%
ディスク制御プロセッサ2	0%	100%	0%	0%
ディスク制御プロセッサ3	0%	0%	100%	0%
ファイル制御メモリ	4GB	5GB	7GB	2GB
キャッシュメモリ	64GB	96GB	96GB	0GB
ホストインターフェース部	H00 H01 H02 H03	H04 H05 H06 H07	H08 H09 H0A H0B	H0C
プロセッサ間通信接続部	40%	25%	35%	0%
ディスクドライブ	400GB/RAID5(0R81 2 3 4)/7.5kRPM 400GB/RAID5(0R85 6 7 8)/7.5kRPM 200GB/RAID5(0R89 10 11 12)/7.5kRPM 200GB/RAID5(0R813 14 15 16)/7.5kRPM	300GB/RAID1(0R811 12)/15kRPM 300GB/RAID1(0R813 14)/15kRPM 300GB/RAID1(0R815 16)/15kRPM	100GB/RAID1(0R811 12)/7.5kRPM 300GB/RAID1(0R813 14)/7.5kRPM 300GB/RAID1(0R815 16)/7.5kRPM 100GB/RAID1(0R817 18)/7.5kRPM	

STORAGE HAVING LOGICAL PARTITIONING CAPABILITY AND SYSTEMS WHICH INCLUDE THE STORAGE

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a storage which is connected to a network and used, in particular to a NAS.

Description of the Related Art

As a method of improving performance of an information processing system, the number of computers provided in the information processing system is simply increased. However, if a large number of computers are used, considerable time and labor are required for supervision of the respective computers, and a total area for installing the computers and total power consumed by the computers increase considerably. In order to solve this problem, there has been devised a technique for setting a high performance computer, logically partitioning resources of the computer into plural sections, and using the respective partitioned sections as a virtual computer independently. This is called logical partitioning (LPAR) of a computer. For example, an example of such a logical partitioning technique is disclosed in JP-A-2003-157177 (corresponding US Patent Publication No. 2003/0097393).

By virtualizing one computer look as if it is functioning as a large number of computers according to the logical

partitioning, a flexible operation becomes possible. For example, various operating systems can be used on the respective virtual computers freely and boot up and shutdown or failure management can be performed independently for each virtual computer. In addition, the number of physical machines is small, which is advantageous in terms of system management, an installation area for the machines, power consumption thereof, and the like. However, in the LPAR in the conventional computer, although resources such as a processor and a memory in the computer are logically partitioned and allocated to the respective virtual computers, concerning a storage connected to the computer, a storage area provided in the storage is simply partitioned and the partitioned storage areas are allocated to the virtual computers, respectively. Nothing is further taken into account specifically.

On the other hand, as a form of using a storage, other than a form in which the storage is directly connected to one computer (hereinafter referred to as "host" in some cases) and used, there is a form in which the storage is shared by plural computers via a network. In the latter form, in particular, a storage which has an interface as a form of a file system, that is, to which file access is possible from the computers, is called a network attached storage (hereinafter referred to as "NAS).

Data is exchanged between the NAS and the hosts by a form

of file having a name and a structure which are recognized by an operating system running on the host. Therefore, in addition to a disk drive for storing data and a control unit therefor, the NAS has a processor and a memory, which are used for translating file input/output to and from the host into data input/output to and from the disk drive.

SUMMARY OF THE INVENTION

The NAS is originally based upon an idea that it is more advantageous to provide a storage shared over a network than to set a storage individually for plural hosts. Thus, the NAS itself has to perform control for sharing a storage area or the like among the plural hosts. In addition, when a certain host reads and writes a large quantity of data, almost the entire processing ability of the NAS is consumed, and ability for inputting and outputting data to and from the other hosts declines. Moreover, when data destruction or failure of the NAS occurs due to operation mistake or the like of a certain host, the data destruction or failure may affect data used by the other hosts.

It is an object of the present invention to reduce control for sharing when the NAS is shared by plural hosts and eliminate mutual interference among the hosts so as to guarantee ability of input/output and localize data destruction or failure. In addition, it is another object of the present invention to

improve a usability of resources such as a processor and a memory in the NAS.

In order to attain the above-mentioned objects, logical partitioning is performed in the NAS. More specifically, there is provided a storage which is connected to a network and receives file access, in which resources held by the storage, for example, disk drives, interfaces with the network, processors controlling file access, and the like are logically partitioned by a control unit provided in the storage to enable respective logical partitions (virtual storages) to operate independently.

Further, it is also possible that the control unit has plural processors and these processors divides logical partitioning or performs logical partitioning as a whole.

Moreover, the control unit may have a supervising terminal to input information necessary for logical partitioning from this supervising terminal. In this case, it is also possible that only an access characteristic of a computer using the storage are inputted to the supervising terminal, and the supervising terminal calculates information necessary for logical partitioning from the access characteristic and communicates the information to the storage.

Moreover, a computer using the storage may also function as a supervising terminal.

BRIEF DESCRIPTION OF THE DRAWINGS

In the accompanying drawings:

Fig. 1 is a diagram showing an example of a structure of a NAS;

Fig. 2 is a diagram showing an example of allocation of a file control memory and a cache memory;

Fig. 3 is a diagram showing an example of allocation of a file control processor and a disk control processor;

Fig. 4 is a diagram showing an example of allocation of an inter-processor communication unit;

Fig. 5 is a diagram showing an example of allocation of groups of disk drives;

Fig. 6 is a diagram showing a concept of user authentication of a host system according to logical partitioning of a host interface unit and the file control processor;

Fig. 7 is a diagram showing an example of a structure of a NAS;

Fig. 8 is a diagram showing an example of a setting input screen for logical partitioning of the NAS; and

Fig. 9 is a diagram showing an example of logical partitioning information of respective resources of the NAS.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Embodiments of the present invention will be hereinafter described with reference to the accompanying drawings. Note

that it is needless to mention that the present invention is not limited to descriptions of the embodiments described below.

Fig. 1 is a diagram showing an example of an embodiment of a NAS to which the present invention is applied. The NAS includes: two host interface units 100 which are connected to a host (hereinafter also referred to as "host system"); three file control processors 101 which translate an input/output request by a unit of file from the host into an input/output request of data by a unit of block; two file control memories 102 which store information necessary for translating an input/output request by a form of file into an input/output request of data by a unit of block (hereinafter also referred to as "translation control data"); four groups of disk drives 107 serving as storage media; three disk control processors 104 which control input/output of data to and from the groups of disk drives 107; two cache memories 106 which temporarily store data inputted to or outputted from the groups of disk drives 107; two disk interface units 106 which connect the disk control processors 104 and the groups of disk drives 107; and two inter-processor communication units 103 which connect the disk control processors 104 and the file control processors 101.

Here, it is also possible that plural disk drives are included in the groups of disk drives 107 and each group takes a RAID configuration. In addition, the "block" is a

predetermined unit which is used when the disk control processors 104 store data in disk drives. In general, 512B is adopted as the block. In addition, the "translation control data" is information indicating a correspondence relation between a file name used in a file system and a position of a file from its top, and the block. In general, the translation control data is expressed as a link structure such as I-Node or a data structure such as an address translation table.

Note that, in this specification, identical devices are denoted by identical reference numerals, and alphabets such as "a" and "b" are attached to the identical reference numerals when the identical devices are distinguished. In addition, the above-mentioned numbers of the respective devices are only examples and do not limit the present invention.

In Fig. 1, an input/output request by a unit of file, which is sent from the host system to the NAS, is processed in the NAS as described below.

First of all, a request for starting file reference (open) designating a file name is sent to the NAS from the host. Next, an actual input/output request for data is sent, and finally, a request for ending file reference (close) is sent.

These requests are received by any one of the host interface units 100 and transferred to any one of the file control processors 101. The file control processor 101 checks the file name requested by the host with reference to the translation

control data stored in the file control memories 102, records the file name started to be used, and translates a data input/output request for the file into a data input/output request to the groups of disk drives 107 in which the data is stored.

The translated data input/output request is sent to any one of the disk control processors 104 via any one of the inter-processor communication units 103. In addition, in the case in which necessary translation control data is not stored in the file control memories 102, the file control processor 101 requests translation control data stored in a predetermined storage area of the groups of disk drives 107 from any one of the disk control processors 104 via the inter-processor communication unit 103.

With respect to the data input/output request (including a request for translation control data) received from the any one of the file control processors 101 via the inter-processor communication unit 103, the disk control processor 104 checks if the data is stored in any one of the cache memories 105. When the data is stored in any one of the cache memories 105, the disk control processor 104 applies writing or reading of the requested data to the cache memory 105.

Thereafter, in the case of writing, the disk control processor 104 returns a result to the effect that writing is completed, or in the case of reading, returns a result to the

effect that reading is completed together with the read-out data to the file control processor 101, which sent the input/output request, via the inter-processor communication unit 103. The file control processor 101 processes the returned result and data and sends the result (data, a report on completion of processing, etc.) to the host system, which sent the input/output request to the NAS, via the host interface unit 100 which received the input/output request.

On the other hand, in the case in which the data requested by the file control processor 101 is not stored in all the cache memories 105, the disk control processor 104 specifies in which portion of the groups of disk drives 107 the requested data is stored, reads out the data from the portion of the groups of disk drives 107 via one of the disk interface units 106, and stores the data in one of the cache memories 105.

Thereafter, the disk control processor 104 applies reading or writing of the requested data to the cache memory 105 in which the data is stored. The subsequent processing is the same as the processing described above.

Note that the data stored in the cache memory 105 is written back to the groups of disk drives 107, for example, when a fixed time has elapsed or when a free space of the cache memory 105 becomes insufficient.

In this embodiment, for example, the above-mentioned kinds of processing are performed independently from each other

according to a unit (logical partition) partitioned by a logical partition boundary as shown in Fig. 1. When the host interface units 100, the file control processors 101, the file control memories 102, the inter-processor communication units 103, the disk control processors 104, the cache memories 105, the disk interface units 106, and the groups of disk drives 107, which are physical resources for processing allocated to the respective logical partitions, are allocated to each logical partition once, the devices are used solely for processing of the logical partition. More specifically, a file control processor 101a and a disk control processor 104c, which are allocated to different logical partitions in Fig. 1, do not exchange the input/output request of data as described above.

In addition, resources (e.g., file control memory 102b), which are shown across the logical partition boundaries in Fig. 1, are used for each logical partition boundary with a capacity or the like thereof logically partitioned at a rate allocated in advance. In this way, the logical partitions operate as virtual NASs independent from each other.

The processing for partitioning and allocating the physical resources to the respective logical partitions is actually executed by the file control processors 101 and the disk control processors 104. As a method of controlling logical partitioning, two methods described blow are conceivable.

In a first method, the file control processors 101 and

the disk control processors 104 divide the control for the logical partitioning of several physical resources and control the logical partitioning in association with each other as a whole.

For example, the file control processors 101 perform processing for allocating the host interface units 100, the file control processors 101, and the file control memories 102. This processing is hereinafter referred to as "file control hypervisor".

In addition, the disk control processors 104 perform processing for allocating the inter-processor communication units 103, the disk control processors 104, the cache memories 105, the disk interface units 106, and the groups of disk drives 107. This processing is hereinafter referred to as "disk control hypervisor". The file control hypervisor executed by the file control processors 101 and the disk control hypervisor executed by the disk control processors 104 cooperate with each other to perform the each allocation processing. Details of cooperation will be described later. Note that the file control hypervisor may be executed by any one of the file control processors 101, for example, the file control processor 101a or may be executed by plural file control processors 101, for example, the file control processors 101a and 101b. This is also true for the disk control hypervisor.

In a second method, the two kinds of control processors

cooperate to control logical partitioning of all the physical resources. More specifically, all the file control processors 101a to 101c and the disk control processors 104a to 104c perform processing for allocating logical partitions of all the resources of the NAS (integrated hypervisor).

More specifically, for example, hypervisor operating on each processor realizes the logical partitioning as described below.

First, the hypervisor makes resources other than IO processing resources in a logical partition, to which each processor is allocated, invisible for basic IO processing software (BIOS) running on the processor. For example, in Fig. 1, the file control processor 101a is physically connected to the host interface unit 100b. However, in the case in which the logical partition is set to be partitioned by an illustrated dotted line, the hypervisor makes the host interface unit 100b invisible.

More specifically, in the case in which a privileged instruction for checking resources connected to the processor and available resources is executed in a BIOS, the hypervisor generates interrupt in terms of software according to execution of the privileged instruction to shift the execution to the hypervisor. The hypervisor checks resources allocated to a logical partition to which the processor belongs, sets a result of the privileged instruction such that only resources allocated

to the logical partition are visible, and returns to the BIOS in which the interrupt was generated.

In this way, each processor handles only resources of a logical partition to which the processor belongs, and separation of logical partitions is realized.

There are two kinds of memories, the host interface units 100 and the inter-processor communication units 103, which are provided with plural communication channels, among the resources. In such a case, it is sufficient to control an amount of resources to be shown to processors in the respective logical partitions (in the case of the memory, capacities of the memory in a start physical address and an end physical address, and in the case of the communication channel, the number of channels indicated by a set of physical numbers of channels).

In addition, concerning the processor itself, in the case in which each processor is allocated to one logical partition completely, it is sufficient to occupy the processor with the logical partition.

On the other hand, it is also conceivable to allocate one certain processor to two or more logical partitions and cause the logical partitions to share the processor by determining their sharing ratios. In such a case, it is conceivable to implement timer interrupt in each processor in terms of hardware to make arrangement such that the hypervisor is started up at each fixed time by the timer interrupt.

The hypervisor started up by the timer interrupt measures to which extent processing of a logical partition has been performed by the processor, determines a logical partition to be processed next in accordance with the predetermined sharing ratios, and shifts the execution of the processor to processing of the logical partition. In this way, one processor can be partitioned at predetermined ratios and allocated to two or more logical partitions.

Note that, as a method of realizing the hypervisor, other than the above-mentioned example, for example, it is also possible to provide special purpose hardware which is connected to the respective processors and performs resource supervision, a special purpose co-processor which is controlled by a small-sized micro program, and the like to realize the control of logical partitioning.

In addition, information on logical partitioning, for example, information designating a processor, a memory, a communication unit, and the like used in a logical partition 1, is stored in any one or more of the file control memories 102, the cache memories 105, the disk drives of the groups of disk drives 107, or the other storages. Each kind of hypervisor reads out the information to thereby apply designation of logical partitioning to the BIOS or the like. Note that this information is set via a supervising terminal to be described later.

Concerning the hypervisor realized as described above,

in the case in which the file control hypervisor is operated by the file control processor and the disk control hypervisor is operated by the disk control processor, the file control hypervisor performs the processing for allocating the host interface units 100, the file control processors 101, and the file control memories 102, the disk control hypervisor performs the processing for allocating the inter-processor communication units 103, the disk control processors 104, the cache memories 105, the disk interface units 106, and the groups of disk drives 107, and the two kinds of hypervisor are associated with each other.

More specifically, in designating logical partitioning with a supervising terminal to be described later, designation applied to logical partitioning with respect to the file control hypervisor and designation applied to logical partitioning with respect to the disk control hypervisor are performed in association with each other. Alternatively, depending upon a manner of designation with respect to logical partitioning, as described later, the file control hypervisor and the disk control hypervisor are adapted to automatically make adjustment each other such that a logical partition conforms to a request for the designated logical partitioning.

In the case of integrated hypervisor, for example, respective processors starting up the integrated hypervisor share information on allocation of all resources to respective

logical partitions, the hypervisor determines resources to be used by a processor, which has started up the hypervisor, with reference to the allocation information and executes allocation processing.

Next, a specific example of allocation of the resources of the NAS to logical partitions will be described. The allocation of the resources will be hereinafter described with the case in which the NAS shown in Fig. 1 is logically partitioned into two logical partitions (logical partition 1 and logical partition 2) as an example. However, any number of logical partitions will do. In addition, in the following description, the file control hypervisor and the disk control hypervisor are described as performing logical partitioning in association with each other. However, logical partitioning may be performed by the integrated hypervisor. Moreover, as a matter of expression, if hypervisor performs logical partitioning, processing by the hypervisor is actually executed by a processor which performs processing of each hypervisor.

Fig. 2 is a diagram showing an example of allocation of the file control memory 102 and the cache memory 105 to logical partitions. For example, when a request of a host system using the logical partition 1 attaches importance to a read data transfer rate, the disk control hypervisor increases an amount of allocation of the cache memory 105 to the logical partition 1 such that requested data is stored in the cache memory 105

as much as possible. In this case, an amount of allocation of the file control memory 102 to the logical partition 1 may be small.

In association with the above, the file control hypervisor decreases the amount of allocation of the file control memory 102 to the logical partition 1 to allocate a larger storage capacity to the logical partition 2. Consequently, it is possible to improve utilization of the file control memory 102 in the NAS as a whole.

On the other hand, if a request of the host system using the logical partition 1 attaches importance to a response time, the file control hypervisor allocates a larger storage capacity of the file control memory 102 to the logical partition 1 such that translation control data is stored in the file control memory 102 as much as possible. In this case, the amount of the cache memory 105 allocated to the logical partition 1 may be small. Consequently, the disk control hypervisor can allocate a larger capacity of the cache memory 105 to the logical partition 2, and it is possible to improve utilization of the cache memory 105 in the NAS as a whole.

In addition, in the case in which an input/output request of the host system using the logical partition 1 mainly concerns random access to data scattered in a large area of the groups of disk drives 107 provided in the NAS, it is difficult to store all pieces of information on the access scattered in a large

area even if large capacities of the file control memory 102 and the cache memory 105 is allocated to the logical partition 1. Thus, an effect of the allocation is small. Therefore, in such a case, the file control hypervisor and the disk control hypervisor reduce the amounts of allocation of the file control memory 102 and the cache memory 105 to the logical partition 1 and allocate a large storage capacity to the logical partition 2 which is the other logical partition to thereby improve a utilization of the cache memory 105 and the like.

Conversely, in the case in which an input/output request of the host system using the logical section 1 mainly concerns sequential access to data stored in continuous ranges of the groups of disk drives 107 provided in the NAS, it is possible to specify in advance information necessary for access and data to be read in advance in the NAS itself. Therefore, it is conceivable that the file control hypervisor and the disk control hypervisor perform allocation processing to increase amounts of allocation of the file control memory 102 and the cache memory 105 to the logical partition 1 such that the information and the data can be stored in the file control memory 102 and the cache memory 105 sufficiently.

Fig. 3 is a diagram showing an example of allocation of the file control processor 101 and the disk control processor 104 to logic partitions.

In the case in which an input/output request from a host

system using the logical partition 1 is a request for a small number of large file accesses, an amount of processing for translating file input/output executed by the file control processor 101 into data input/output is not so large. Therefore, an amount of allocation of the file control processor 101 to the logical partition 1 may be small.

In this case, the file control hypervisor reduces the amount of allocation of the file control processor 101 to the logical partition 1 (more specifically, an occupation ratio of the processor) and allocates relatively a larger amount of processor resources than that of the logical partition 1 to the logical partition 2 which is the other logical partition. Consequently, it becomes possible to improve a utilization of the file control processor 101 provided in the NAS.

In addition, in this case, since an amount of data of a file is large, an amount of processing of data input/output executed by the disk control processor 104 allocated to the logical partition 1 increases. Therefore, the disk control hypervisor increases an amount of allocation of the disk control processor 104 to the logical partition 1.

Further, in the case in which an input/output request of the host system using the logical partition 1 is a request for a large number of small file accesses, an amount of processing for translating file input/output executed by the file control processor 101 into data input/output increases. Thus, the file

control hypervisor increases an amount of allocation of the file control processor 101 to the logical partition 1.

In this case, since an amount of data of a file is small in itself, an amount of processing for data input/output executed by the disk control processor 104 allocated to the logical partition 1 is not so large. Thus, the disk control hypervisor reduces the amount of allocation of the disk control processor 104 to the logical partition 1 and increases an amount of allocation of the disk control processor 104 to the logical partition 2. Consequently, it becomes possible to improve a utilizaiton of the disk control processor 104 in the NAS.

Moreover, in the case in which the host system using the logical partition 1 does not require a high performance NAS, the file control hypervisor and the disk control hypervisor perform allocation processing so as to reduce the amounts of allocation of the file control processor 101 and the disk control processor 104 to the logical partition 1. Conversely, in the case in which the host system using the logical partition 1 requires a high performance NAS, the file control hypervisor and the disk control hypervisor perform allocation processing so as to increase the amounts of allocation of the file control processor 101 and the disk control processor 104 to the logical partition 1.

Fig. 4 is a diagram showing an example of allocation of the inter-processor communication unit 103 to logical

partitions. In the case in which an input/output request from a host system using the logical partition 1 is a request for a large file sequential access, the disk control hypervisor performs allocation processing so as to increase an amount of allocation of the inter-processor communication unit 103 to the logical partition 1 (more specifically, a communication bandwidth) and preserve data communication ability between the file control processor 101 and the disk control processor 104 (in other words, data communication ability between the host interface unit 100 to the cache memory 105).

In addition, if an input/output request of the host system using the logical partition 1 is a request for a small file sequential access, the amount of allocation of the inter-processor communication unit 103 to the logical partition 1 may not be large. Moreover, if a request from the host system is a request for a random access, performance of a virtual NAS of the logical partition 1 from the viewpoint of the host system is not significantly affected by the amount of allocation of the inter-processor communication unit 103 to the logical partition 1. Therefore, in these cases, the disk control hypervisor performs allocation processing so as to reduce the amount of allocation of the inter-processor communication unit 103 to the logical partition 1 and increase allocation thereof to the other logical partition (here, the logical partition 2) and improve utilization of the inter-processor communication

unit 103 in the NAS.

Fig. 5 is a diagram showing an example of allocation of the groups of disk drives 107 to logical partitions. If a host system using the logical partition 1 prefers large storage capacity, the disk control hypervisor allocates the groups of disk drives 701, which has a RAID 5 configuration with a high storage capacity efficiency (in Fig. 7, while the number of disk drives in which data is stored is three, the number of disk drives in which parity is stored is one, and a capacity efficiency is 75%), to the logical partition 1. In this case, a speed of rotation of a disk drive may not be so fast, for example, 7,500 rpm.

On the other hand, if the host system using the logical partition 1 prefers good access performance, the disk control hypervisor allocates the group of disk drives 701, which has a RAID1 configuration allowing accessibility to be improved (in Fig. 7, since identical data is duplicated and stored in two disk drives, a storage capacity efficiency is 50%, but since both the two disk drives can be used for the identical data, total accessibility is twice as high as that of one disk drive), to the logical partition 1. Note that, in this case, taking into account speeds of rotation of disk drives included in the group of disk drives 107 as well, the disk control hypervisor may allocate the group of disk drives 107 having a disk drive of a high rotation speed, for example, 15,000 rpm among the

groups of disk drives 107 of the same RAID1 configuration to the logical partition 1.

Note that allocation of the host interface unit 100 to logical partitions is performed by the file control hypervisor according to performance required by a host system using the each logical partition. More specifically, in the case in which the performance required by the host system is high, the file control hypervisor allocates a large amount to a logical partition used by the host system, that is, high ability of communication with the host system (communication bandwidth, etc.). On the other hand, in the case in which the performance required by the host system is low or, in particular, there is no request, it is conceivable that the file control hypervisor allocates a small amount to a logical partition used by the host system, that is, low ability of communication with the host system and improve efficiency of the entire NAS.

Moreover, by logically partitioning to use one NAS as in this embodiment, it is possible to perform user authentication for a host system in the NAS independently by each logical partition. Fig. 6 is a diagram showing an example of the user authentication.

In this figure, a user A with an identifier (hereinafter referred to as "ID") "abc" and a user B with an ID "def" use a host system A which uses a logical partition 1, and a user C with an ID "ghi" and a user D with an ID "abc" use a host

system B which uses a logical partition 2. In this case, the user A of the host system A and the user D of the host system B have the same ID "abc". Thus, in order to distinguish the user B and the user D in a conventional NAS, it has been necessary to perform special processing such as giving IDs to host systems or a group of host systems and combining the host ID and IDs of users to distinguish the users.

However, in this embodiment, the host interface unit 100 and the file control processor 101 are logically partitioned, and the each logical partition operates as the host interface units 100 and the file control processors 101 of separate individual virtual NASs. Thus, user authentication is also performed independently for each logical partition. In other words, the user A and the user D having the identical ID "abc" in Fig. 8 are authenticated in the separate logical partitions respectively. Therefore, the user A and the user D are distinguished naturally, and no special processing is required in order to distinguish the users. In other words, as long as logical partitions are different, an identical ID can be given to plural users without performing special processing.

Moreover, resources of the host interface unit 100 and the file control processor 101 allocated to each logical partitions are never used in the other logical partitions. Thus, even if a user of a certain logical partition performs a large quantity of data access, users of the other logical partitions

are never affected by that data access.

Next, a second embodiment will be described. A NAS of this embodiment includes processors of one type and memories of one type, in which the file control processors 101 and the disk control processors 104 of the NAS are integrated and the file control memories 102 and the cache memories 105 of the NAS are integrated.

Fig. 7 is a diagram showing an example of a structure of the second embodiment. In Fig. 7, integrated control processors 901 are processors in which the file control processors 101 and the disk control processors 104 are integrated, and integrated control memories 902 are memories in which the file control memories 102 and the cache memories 105 are integrated.

Compared with the previously-mentioned embodiment (Fig. 1), in this embodiment, the inter-processor communication units 103 become unnecessary and the system structure is simplified. In Fig. 7, both of processing for translating an input/output request by a unit of file from a host system into an input/output request of data and control of input/output of data between the disk interface units 106 and the groups of disk drives 107 are performed by the integrated processors 901. In addition, translation control information and data of the groups of disk drives 107 are stored in the integrated control memories 902. Structures and operations of the other portions in Fig. 7 are

the same as those in Fig. 1.

In Fig. 7, as in Fig. 1, for example, processing is performed independently in logical partitions separated by a logical partition boundary as shown in Fig. 7. When the host interface units 100, the integrated control processors 901, the integrated control memories 902, the disk interface units 106, and the groups of disk drives 107, all which are physical resources for processing allocated to the respective logical partitions, are allocated to each logical partition once, the devices are used solely for processing of the logical partition. In this way, the respective logical partitions operate as virtual NASs independent from each other.

The processing for logical partitioning and allocating the physical resources to the respective logical partitions is actually executed by the integrated control processors 901. The integrated control processors 901 perform control of the previously-mentioned integrated control hypervisor.

Fig. 8 is a diagram showing an example of a setting input screen of a supervising terminal for inputting setting of logical partitioning of a NAS. Such a setting input screen is used in both the first embodiment and the second embodiment. A supervisor or the like inputs setting for logical partitioning as shown in Fig. 8, and contents of the input of setting are notified to the NAS, whereby hypervisor operating in the NAS logically partitions the respective resources of the NAS. More

specifically, contents of the input of setting are stored in a certain storage area provided in the NAS, and the hypervisor performs logical partitioning in accordance with the stored information.

More specifically, such a supervising terminal may be provided in a host system which has connection to the NAS through a network. Or, the supervising terminal may be realized by a console device connected to the NAS by a special line. Such a control device can be realized by an input device such as a keyboard and a display device such as a display.

Information inputted by the supervisor or the like is transferred to the NAS from the host system or the console device using a special purpose protocol or a general purpose protocol. The NAS has an interface for receiving the information (e.g., the host interface units 100 or a special purpose interface).

The setting input screen shown in Fig. 8 will be hereinafter described in detail. The screen includes a field in which the number of partitions of logical partitioning is inputted. The supervisor or the like inputs an intended number of partitions of logical partitioning in this field first (3 in the example of Fig. 8). When the supervisor or the like inputs the number of partitions of logical partitioning, logical partitions corresponding to the number are displayed on the screen for each physical resource, and an initial value of resource allocation to each logical partition is displayed.

Thereafter, the supervisor or the like inputs allocations of the resources such as processors or memories while looking at the screen. In this case, for example, in a part where allocation of the file control processor 101 and the disk control processor 104 is designated, if display is adapted such that ways of allocation of the respective processors to the respective logical partitions can be set in association with each other as shown in the figure (in the figure, both the processors are displayed side by side such that a relation between the processors can be easily seen), it becomes easy to input the setting for allocation control as described above with reference to Fig. 5.

In addition, for example, as shown in Fig. 8, not only knobs for setting amounts of allocation of the file control processor 101 and the disk control processor 104 to the logical partitions individually (icons which can be selected by a pointing device) but also knobs with which the amounts of allocation can be set in association with each other are prepared. Similarly, concerning the file control memory 102 and the cache memory 105, amounts of allocation of the memories to the respective logical partitions are displayed in association with each other, and knobs for individual setting and associated setting are prepared.

In the example of Fig. 8, concerning the inter-processor communication unit 103, the supervisor or the like inputs

percentages of allocation of entire data transfer ability between the file control processor 101 and the disk control processor 104 (the host interface unit 100 and the cache memory 105) to the respective logical partitions.

In addition, concerning the host interface unit 100, the supervisor or the like inputs information on allocation by specifying a connection port of a network for physically connecting the host system rather than according to percentages of the resources. However, a method of inputting the information simply according to the percentages of the resources may be adopted. Moreover, concerning the group of disk drives 107, the supervisor or the like performs allocation of the resources by setting capacities, RAID constitutions, and performance (speed of rotations) of physical disk drives to be allocated to the respective logical partitions.

The above-mentioned methods of input of setting for resource allocation are examples. Other than these methods of input, allocation of resources may be inputted independently as numerical values or may be inputted automatically to some extent. For example, the supervisor or the like inputs characteristics of access which a host system requires of a certain logical partition (random or sequential, an average data length per one transfer, a minimum data transfer rate, a maximum response time, etc.) via a supervising terminal, whereby the supervisor or the like selects a set of parameters

meeting the inputted characteristics with hypervisor, from several sets of parameters prepared in advance, more specifically, from sets of parameters having the characteristics shown in Figs. 2 to 5.

Consequently, for example, in the case in which the supervisor or the like designates sequential access, the supervisor or the like selects a set of parameters corresponding to sequential (parameters corresponding to sequential shown in Figs. 2 and 4) with hypervisor from the sets of parameters prepared in advance and sets the parameters to perform processing of logical partitioning.

More specifically, when the supervisor or the like wishes to create three logical partitions, namely, a logical partition suitable for a large file access, a logical partition suitable for small file access, and a moderate size file access, the supervisor or the like prepares partitioning parameters for a logical partition 1, in which an amount of allocation of a file control processor is smaller than an amount of allocation of a disk control processor, a logical partition 2, in which an amount of allocation of a file control processor is larger than an amount of allocation of a disk control processor, and a logical partition 3, in which amounts of allocation of a file control processor and a disk control processor are comparable, as described in the example of allocation setting of a file control processor and a disk control processor in Fig. 8, in

an area referred to by the hypervisor (in a storage area in which information on allocation of resources to logical partitions described later in Fig. 9 is stored).

In actually performing allocation setting, the supervisor or the like simply designates characteristics which are requested of logical partitions such as the three logical partitions consisting of the logical partition suitable for large file access, the logical partition for small file access, and the logical partition for moderate size file access. Then, the hypervisor automatically selects parameters for allocation corresponding to the designation.

Consequently, the supervisor or the like can easily designate logical partitions having desired performance and characteristics.

In addition, the supervisor or the like is required to make sure such that resources sufficient for the each logical partition allowing it to operate correctly are always allocated by the input of allocation setting. For example, an amount of allocation of the file control processor or the disk control processor cannot be set to zero. At this point, in the automatic setting as described above, it is also possible that a lower limit is set for an amount of allocated resources in advance such that this lower limit is complied with automatically. In addition, in the example of input as shown in Fig. 8, it is also possible that lower limits of amounts of allocation of

the respective resources are defined in advance in the NAS and, in the case in which an amount of allocation of a resource below the lower limit for the resource is inputted, an warning is reported or such input of allocation is checked and refused.

Consequently, the supervisor or the like can set logical partitions safely.

Fig. 9 is an example showing information indicating the allocation of the resources to the logical partitions set by the supervisor or the like as described above. A correspondence relation between the physical resources and the logical partitions as shown in Fig. 9 is created on the basis of information that the hypervisor has received from the supervising terminal. More specifically, the respective parts of hypervisor have information on structures of the physical resources of the NAS, allocate the physical resources to the respective logical partitions on the basis of information inputted by the supervisor or the like and the information on the structure, and create the correspondence relation as shown in Fig. 9. Note that, in Fig. 9, the number of items on the vertical axis increases and decreases according to structures of devices provided in the NAS, and the number of logical partitions on the horizontal axis is changed according to designation by the supervisor or the like.

Then, the information on the correspondence relation as shown in Fig. 9 is stored in an area solely used by hypervisor

of any one or more of the file control memories 102, the cache memories 105, the groups of disk drives 107, and the other storage media as described above. The respective parts of hypervisor determine resources, which the respective logical partitions are caused to use, with reference to the stored information and perform processing for allocation of the resources.

In the storage according to the present invention, when the storage is shared by plural hosts, control for the sharing can be reduced, and mutual interference among the hosts can be eliminated to guarantee performance of data input/output and localize data destruction or failure.

In addition, user authentication independent for each host or each group of hosts can be performed. Further, according to the present invention, it is possible to improve utilization of resources such as processors, memories, and storage media in a system.

Moreover, plural virtual NASs can be provided by one NAS. A degree of freedom of an operating system of a host can be improved. It is possible to independently perform operation and stop or failure processing. Thus, the NAS becomes advantageous in terms of system management, an installation area, power consumption, and so on.

Fig. 10 is an example of a form in which an NAS is connected to a host system (host computer). The NAS according to the present invention can also be used in the form of Fig. 10.

In Fig. 10, four host computers 900a to 900d are connected to an NAS 902 by a network 901. A supervising terminal 903 is connected to the NAS 902 by a special line 904. By using the special line 904, the supervising terminal 903 can be connected to the NAS 902 even if no settings are made in the NAS 902 in advance. When the NAS 902 is used for the first time, since no settings are made concerning a network, it is possible that the supervising terminal 903 is connected in the form as shown in Fig. 10.

On the other hand, Fig. 11 is an example of a form in which the supervising terminal 903 is also connected to the NAS 902 through the network 901 without providing a special line between the supervising terminal 903 and the NAS 902. In this case, settings concerning a network have to be made in the NAS 902 in order to communicate with the supervising terminal 903 through the network. If the network 901 is, for example, an IP network, settings for IP addresses of the NAS 902 itself and the supervising terminal 903 and network masks are necessary.

It is possible that such settings concerning a network for communicating with the supervising terminal 903 are performed by, for example, connecting the supervising terminal 903 to the NAS 902 once through a special line in the form as shown in Fig. 10. When the settings for the network are completed, the connection by the special line between the supervising terminal 903 and the NAS 902 can be cancelled and removed, and

the supervising terminal 903 can be connected to the network 901 to change the form to the form of Fig. 11, whereby the NAS 902 can be supervised from the supervising terminal 903 through the network.

In addition, in Fig. 11, another method of performing the settings for the network for communicating with the supervising terminal 903 is to install a very small console unit for only performing the settings for the network in the NAS 902. Fig. 12 is an example of such a console unit. If the console unit as shown in Fig. 12 is provided on an appropriate surface of a housing of the NAS 902, the network settings for communicating with the supervising terminal 903 through the network 901 in the form as shown in Fig. 11 can be performed. If the supervising terminal 903 can be connected once through the network 901, supervising work for the NAS 902 after that can be performed through the supervising terminal 903.

In the NAS according to the present invention, plural virtual NASs operate on one physical NAS. For identifying the respective virtual NASs, settings are made in principle in network connection ports (host channels) of host interface units, which are allocated to the respective virtual NASs (logical partitions), such that the virtual NASs are identified by the network individually. For example, in the case in which the virtual NASs make connection through an IP network, different IP addresses have to be assigned to the respective host channels.

Such identification settings for the network (assigning of IP addresses) can be performed by making connections between the supervising terminal and each of the virtual NASs operating in each logical partition. When the supervising terminal is connected to the physical NAS, for example, by a special line as shown in Fig. 10, it is possible to provide a special switch on the supervising terminal side or the physical NAS side for switching the virtual NAS to be connected to the supervising terminal. For example, it is possible that such a switch is adapted such that currently operating virtual NASs are switched to be connected to the supervising terminal in turn every time the switch is pressed. On the supervising terminal side, it is also possible that such a special switch is substituted with some special sequence of normal key switches on the supervising terminal.

In addition, in the case in which the supervising terminal is connected through the network as shown in Fig. 11, identification settings for supervising (if the network is an IP network, assignments of IP addresses) different from identification settings of the network given to the host channels are performed in the NAS. In this case, first, the supervising terminal makes connection to the NAS using an IP address for supervision. Next, the special switch is provided such that currently operating virtual NASs are switched in turn every time the switch is pressed. Alternatively, it is also possible

to prepare IP addresses for supervision by the number of virtual NASs (the number of logical partitions). In that case, it is unnecessary to provide a special switch, but it is necessary to prepare addresses of the network for supervision by the number of virtual NASs.

WHAT IS CLAIMED IS:

1. A storage to be connected to a network, comprising:

a plurality of interfaces which is connected to the network and receives file access;

a plurality of disk drives; and

a control unit which translates the file access into block access and controls the plurality of disk drives on the basis of the block access,

wherein the control unit logically partitions the plurality of interfaces, the plurality of disk drives and the control unit and causes the partitioned plurality of interfaces, the partitioned plurality of disk drives and the partitioned control unit to operate as a plurality of virtual storages independently.

2. A storage according to claim 1, wherein the control unit further includes a plurality of cache memories, and the plurality of cache memories is logically partitioned and allocated to the respective plurality of virtual storages.

3. A storage according to claim 2, wherein the control unit further includes a first processor, which translates the file access into the block access, and a second processor, which controls the plurality of disk drives on the basis of the block

access, and

wherein the first processor and the second processor are logically partitioned, respectively, and allocated to the respective plurality of virtual storages.

4. A storage according to claim 3, wherein the first processor executes first hypervisor which performs logical partitioning of the plurality of interfaces and the first processor, and

wherein the second processor executes second hypervisor which performs logical partitioning of the plurality of cache memories, the plurality of disk devices and the second processor.

5. A storage according to claim 4, wherein the control unit further includes a plurality of memories which is used by the first processor and a plurality of communication units which connects the first processor and the second processor,

wherein the plurality of memories is logically partitioned by the first hypervisor and the plurality of communication units is logically partitioned by the second hypervisor.

6. A storage according to claim 3, wherein the first processor and the second processor execute hypervisor which performs logical partitioning of the plurality of interfaces,

the first processor, the plurality of cache memories, the second processor, and the plurality of disk drives.

7. A storage according to claim 1, wherein the control unit executes hypervisor which performs logical partitioning of the plurality of interfaces, the control unit, and the plurality of disk drives.

8. A storage according to claim 3 further connected to a supervising terminal,

wherein the control unit performs the logical partitioning on the basis of information inputted from the supervising terminal.

9. A storage according to claim 8, wherein, if information to be inputted to the supervising terminal is information to the effect that a host system using the storage attaches importance to a data transfer rate, an amount of allocation of the plurality of cache memories to a virtual storage to be used by the host system among the plural virtual storages is increased.

10. A storage according to claim 8, wherein, if information to be inputted to the supervising terminal is information to the effect that a host system using the storage

performs random access in a large area, an amount of allocation of the plurality of cache memories to a virtual storage to be used by the host system among the plural virtual storages is reduced.

11. A storage according to claim 5 further connected to a supervising terminal,

wherein the control unit performs the logical partitioning on the basis of information inputted from the supervising terminal.

12. A storage according to claim 11, wherein, if information to be inputted to the supervising terminal is information to the effect that a host system using the storage performs sequential continuous access, an amount of allocation of the plurality of cache memories and the plurality of memories which is used by the first processor to a virtual storage to be used by the host system among the plural virtual storages is increased.

13. A storage according to claim 8, wherein, if information to be inputted to the supervising terminal is information to the effect that a host system using the storage accesses a small number of large files, an amount of allocation of the first processor to a virtual storage to be used by the

host system among the plural virtual storages is reduced, and an amount of allocation of the second processor to the virtual storage is increased.

14. A storage according to claim 8, wherein, if information to be inputted to the supervising terminal is information to the effect that a host system using the storage accesses a large number of small files, an amount of allocation of the first processor to the a virtual storage to be used by the host system among the plural virtual storages is increased, and an amount of allocation of the second processor to the virtual storage is reduced.

15. A storage according to claim 11, wherein information to be inputted to the supervising terminal is information to the effect that a host system using the storage sequentially accesses a large file, an amount of logical allocation of the plurality of communication units to a virtual storage to be used by the host system among the plural virtual storages is reduced.

16. A storage system comprising:

a storage comprising a plurality of interfaces which is connected to the network and receives file access, a plurality of disk drives, and a control unit which translates the file

access into block access and controls the plurality of disk drives on the basis of the block access; and

a supervising terminal which is connected to the storage,

wherein the storage logically partitions the plurality of interfaces, the plurality of disk drives, and the control unit on the basis of information to be inputted to the supervising terminal and operates as plural virtual storages independently.

17. A storage system according to claim 16, wherein information to be inputted to the supervising terminal is information on characteristics of accesses of a computer using the storage, and the storage calculates an amount of logical partitioning of resources provided in the storage on the basis of the information on characteristics of accesses to be inputted to the supervising terminal and performs the logical partitioning using a result of the calculation.

18. A storage to be connected to a network, comprising:

a plurality of interfaces which is connected to the network and receives file access;

a plurality of disk drives; and

a control unit which translates the file access into block access and controls the plurality of disk drives on the basis of the block access,

wherein the control unit further includes a plurality

of cache memories, a first processor, which translates the file access into the block access, a second processor, which controls the plurality of disk drives on the basis of the block access, a plurality of memories which is used by the first processor and a plurality of communication units which connects the first processor and the second processor,

wherein the control unit logically partitions the plurality of cache memories, the first processor, the second processor, the plurality of interfaces, the plurality of disk drives, the plurality of memories, the plurality of communication units and the control unit and causes the partitioned devices to operate as a plurality of virtual storages independently.

ABSTRACT OF THE DISCLOSURE

A storage includes: host interface units; file control processors which receives a file input/output request and translates the file input/output request into a data input/output request; file control memories which store translation control data; groups of disk drives; disk control processors; disk interface units which connect the groups of disk drives and the disk control processors; cache memories; and inter-processor communication units. The storage logically partitions these devices to cause the partitioned devices to operate as two or more virtual NASs.

FIG.1

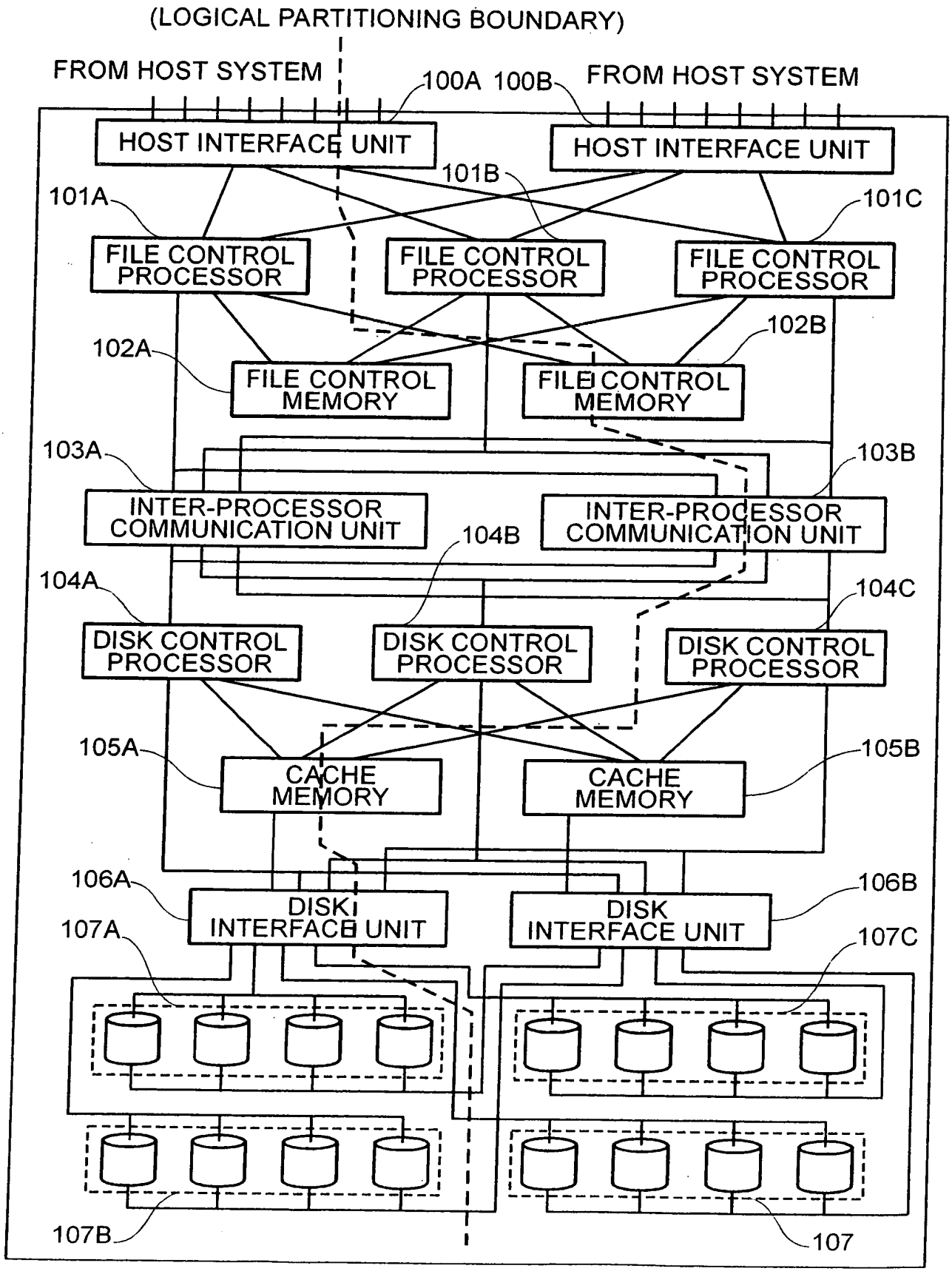
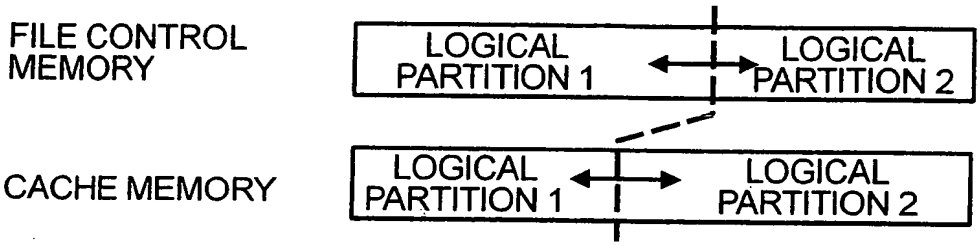
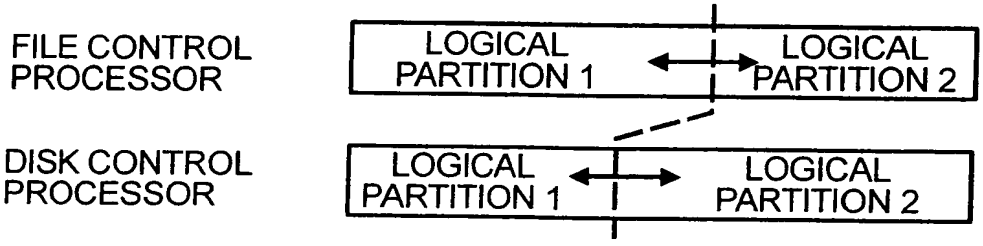

(LOGICAL PARTITIONING BOUNDARY)
FROM HOST SYSTEM
100A
100B
FROM HOST SYSTEM
HOST INTERFACE UNIT
HOST INTERFACE UNIT
101A
101B
101C
FILE CONTROL PROCESSOR
FILE CONTROL PROCESSOR
FILE CONTROL PROCESSOR
102A
102B
FILE CONTROL MEMORY
FILE CONTROL MEMORY
103A
103B
INTER-PROCESSOR COMMUNICATION UNIT
INTER-PROCESSOR COMMUNICATION UNIT
104A
104B
104C
DISK CONTROL PROCESSOR
DISK CONTROL PROCESSOR
DISK CONTROL PROCESSOR
105A
105B
CACHE MEMORY
CACHE MEMORY
106A
106B
DISK INTERFACE UNIT
DISK INTERFACE UNIT
107A
107C
107B
107

FIG.2



	IMPORTANCE IS ATTACHED TO READ DATA TRANSFER RATE	IMPORTANCE IS ATTACHED TO RESPONSE TIME	LARGE AREA RANDOM ACCESS	SEQUENTIAL ACCESS
FILE CONTROL MEMORY	ALLOCATION SMALL	ALLOCATION LARGE	ALLOCATION SMALL	ALLOCATION LARGE
CACHE MEMORY	ALLOCATION LARGE	ALLOCATION SMALL	ALLOCATION SMALL	ALLOCATION LARGE

FIG.3

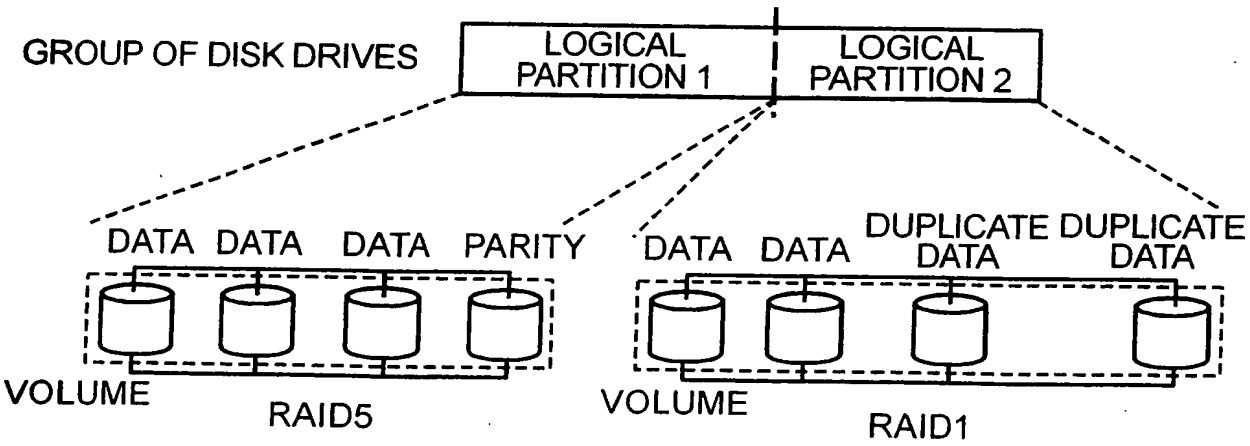


	SMALL NUMBER OF LARGE FILE ACCESS	LARGE NUMBER OF SMALL FILE ACCESS	NOT REQUIRED HIGH PERFOR-MANCE	REQUIRED HIGH PERFOR-MANCE
FILE CONTROL PROCESSOR	ALLOCATION SMALL	ALLOCATION LARGE	ALLOCATION SMALL	ALLOCATION LARGE
DISK CONTROL PROCESSOR	ALLOCATION LARGE	ALLOCATION SMALL	ALLOCATION SMALL	ALLOCATION LARGE

FIG.4

	LARGE FILE SEQUENTIAL ACCESS	SMALL FILE SEQUENTIAL ACCESS	RANDOM ACCESS
INTER- PROCESSOR COMMUNICATION UNIT	ALLOCATION LARGE	ALLOCATION MEDIUM	ALLOCATION SMALL

FIG.5



	CAPACITY PRIORITIZED	ACCESS PERFORMANCE PRIORITIZED
GROUP OF DISK DRIVES	RAID 5 VOLUME ALLOCATION	RAID 1 VOLUME ALLOCATION

FIG.6

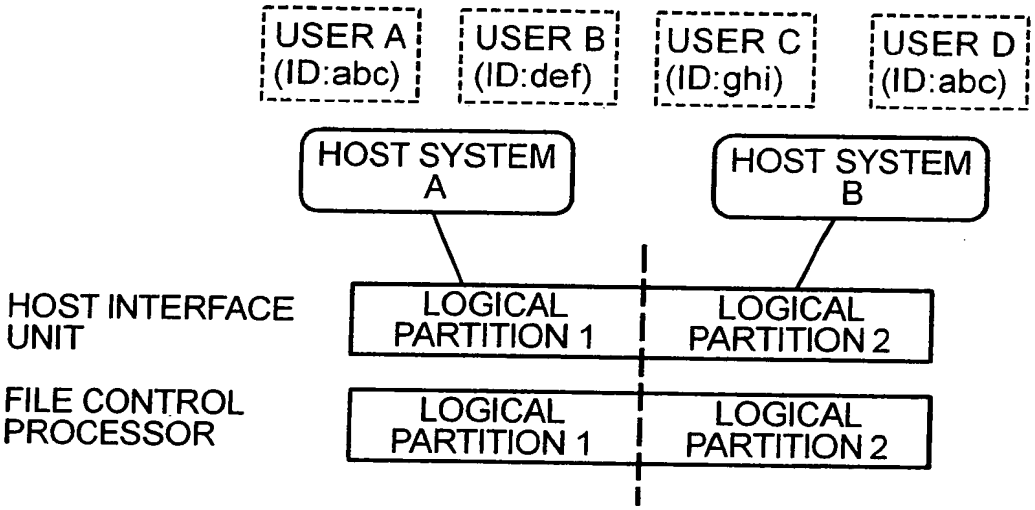

USER A
(ID:abc)
USER B
(ID:def)
USER C
(ID:ghi)
USER D
(ID:abc)
HOST SYSTEM
A
HOST SYSTEM
B
HOST INTERFACE
UNIT
LOGICAL
PARTITION 1
LOGICAL
PARTITION 2
FILE CONTROL
PROCESSOR
LOGICAL
PARTITION 1
LOGICAL
PARTITION 2

FIG.7

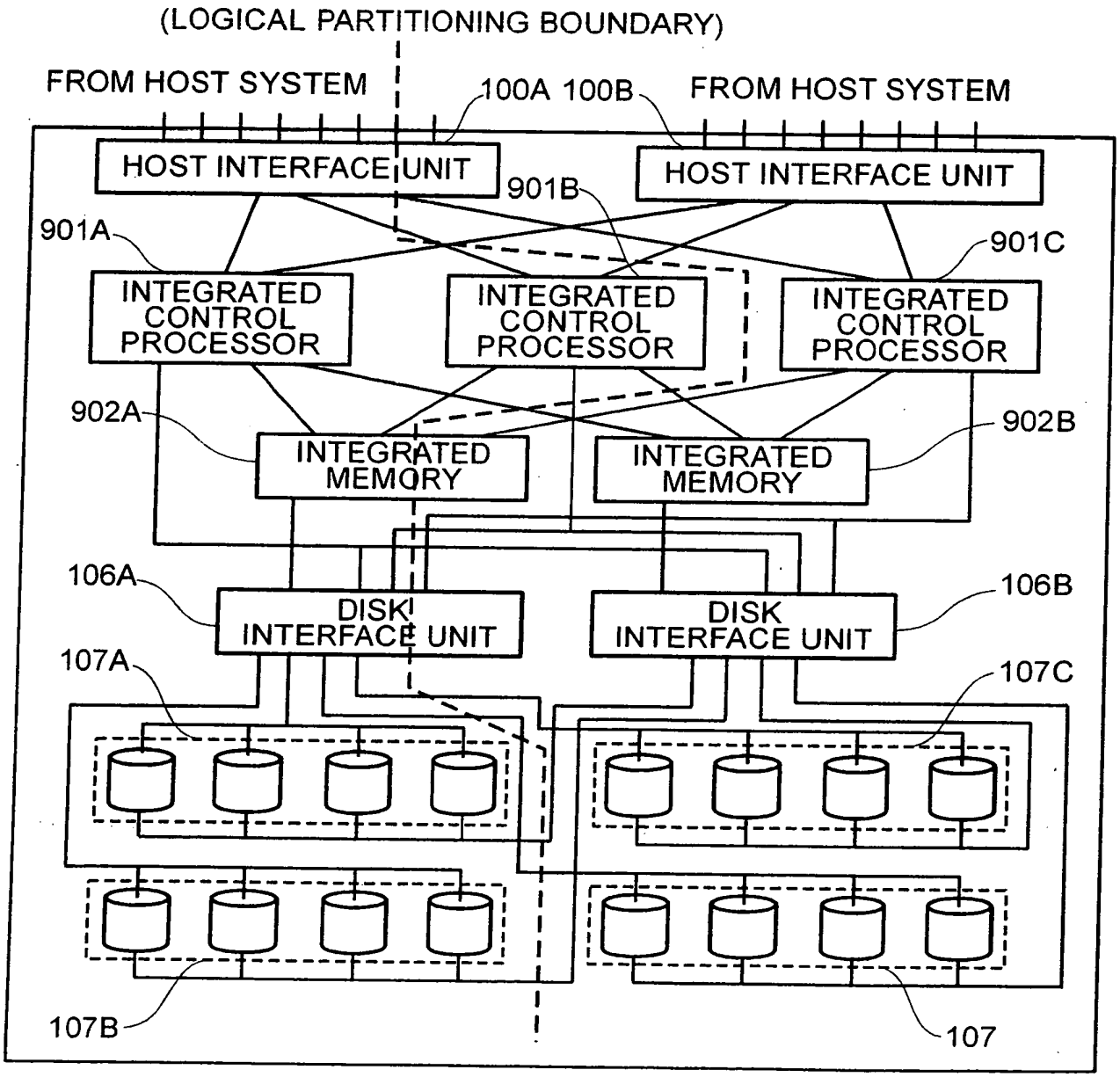

(LOGICAL PARTITIONING BOUNDARY)
FROM HOST SYSTEM
100A 100B
FROM HOST SYSTEM
HOST INTERFACE UNIT
HOST INTERFACE UNIT
901B
901A
901C
INTEGRATED CONTROL PROCESSOR
INTEGRATED CONTROL PROCESSOR
INTEGRATED CONTROL PROCESSOR
902A
902B
INTEGRATED MEMORY
INTEGRATED MEMORY
106A
106B
DISK INTERFACE UNIT
DISK INTERFACE UNIT
107A
107C
107B
107

FIG.8

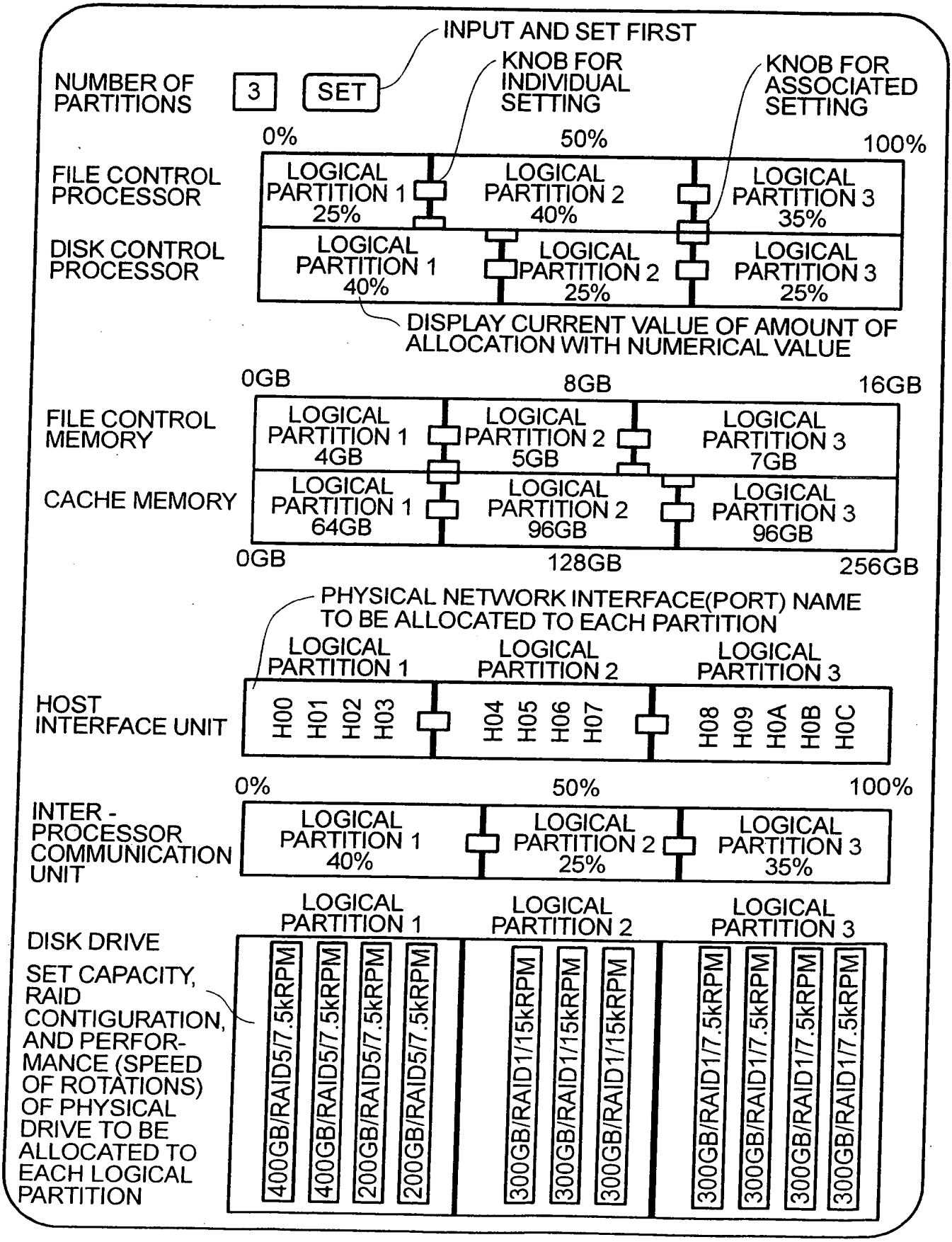

INPUT AND SET FIRST
NUMBER OF PARTITIONS
3
SET
KNOB FOR INDIVIDUAL SETTING
KNOB FOR ASSOCIATED SETTING
0%
50%
100%
FILE CONTROL PROCESSOR
LOGICAL PARTITION 1 25%
LOGICAL PARTITION 2 40%
LOGICAL PARTITION 3 35%
DISK CONTROL PROCESSOR
LOGICAL PARTITION 1 40%
LOGICAL PARTITION 2 25%
LOGICAL PARTITION 3 25%
DISPLAY CURRENT VALUE OF AMOUNT OF ALLOCATION WITH NUMERICAL VALUE
0GB
8GB
16GB
FILE CONTROL MEMORY
LOGICAL PARTITION 1 4GB
LOGICAL PARTITION 2 5GB
LOGICAL PARTITION 3 7GB
CACHE MEMORY
LOGICAL PARTITION 1 64GB
LOGICAL PARTITION 2 96GB
LOGICAL PARTITION 3 96GB
0GB
128GB
256GB
PHYSICAL NETWORK INTERFACE(PORT) NAME TO BE ALLOCATED TO EACH PARTITION
LOGICAL PARTITION 1
LOGICAL PARTITION 2
LOGICAL PARTITION 3
HOST INTERFACE UNIT
H00 H01 H02 H03
H04 H05 H06 H07
H08 H09 H0A H0B H0C
0%
50%
100%
INTER - PROCESSOR COMMUNICATION UNIT
LOGICAL PARTITION 1 40%
LOGICAL PARTITION 2 25%
LOGICAL PARTITION 3 35%
LOGICAL PARTITION 1
LOGICAL PARTITION 2
LOGICAL PARTITION 3
DISK DRIVE
SET CAPACITY, RAID CONTIGURATION, AND PERFOR-MANCE (SPEED OF ROTATIONS) OF PHYSICAL DRIVE TO BE ALLOCATED TO EACH LOGICAL PARTITION
400GB/RAID5/7.5kRPM
400GB/RAID5/7.5kRPM
200GB/RAID5/7.5kRPM
200GB/RAID5/7.5kRPM
300GB/RAID1/15kRPM
300GB/RAID1/15kRPM
300GB/RAID1/15kRPM
300GB/RAID1/7.5kRPM
300GB/RAID1/7.5kRPM
300GB/RAID1/7.5kRPM
300GB/RAID1/7.5kRPM

FIG.9

	LOGICAL PARTITION 1	LOGICAL PARTITION 2	LOGICAL PARTITION 3	NOT ALLOCATED YET
FILE CONTROL PROCESSOR 1	100%	0%	0%	0%
FILE CONTROL PROCESSOR 2	0%	50%	50%	0%
FILE CONTROL PROCESSOR 3	0%	0%	0%	100%
DISK CONTROL PROCESSOR 1	100%	0%	0%	0%
DISK CONTROL PROCESSOR 2	0%	100%	0%	0%
DISK CONTROL PROCESSOR 3	0%	0%	100%	0%
FILE CONTROL MEMORY	4GB	5GB	7GB	2GB
CACHE MEMORY	64GB	96GB	96GB	0GB
HOST INTERFACE UNIT	H00 H01 H02 H03	H04 H05 H06 H07	H08 H09 H0A H0B	H0C
INTER-PROCESSOR COMMUNICATION	40%	25%	35%	0%
DISK DRVE	400GB/RAID5(DR#1.2.3.4)/7.5kRPM 400GB/RAID5(DR#5.6.7.8)/7.5kRPM 200GB/RAID5(DR#9.10.11.12)/7.5kRPM 200GB/RAID5(DR#13.14.15.16)/7.5kRPM	300GB/RAID1(DR#21.22)/15kRPM 300GB/RAID1(DR#23.24)/15kRPM 300GB/RAID1(DR#25.26)/15kRPM	300GB/RAID1(DR#31.32)/7.5kRPM 300GB/RAID1(DR#33.34)/7.5kRPM 300GB/RAID1(DR#35.36)/7.5kRPM 300GB/RAID1(DR#37.38)/7.5kRPM	

FIG.10

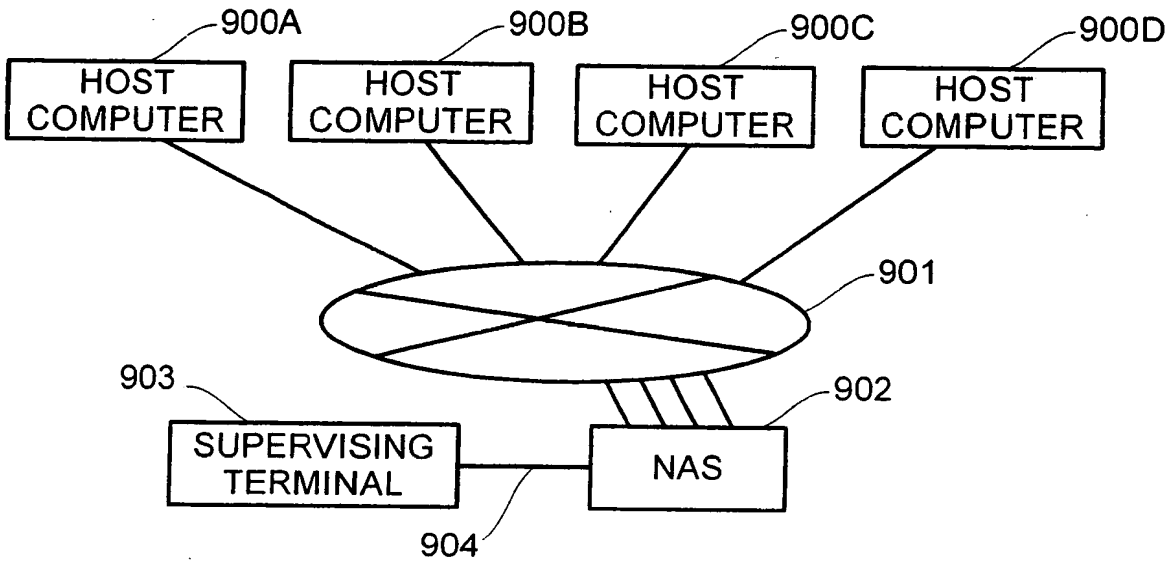

900A
900B
900C
900D
HOST COMPUTER
HOST COMPUTER
HOST COMPUTER
HOST COMPUTER
901
903
902
SUPERVISING TERMINAL
NAS
904

FIG.11

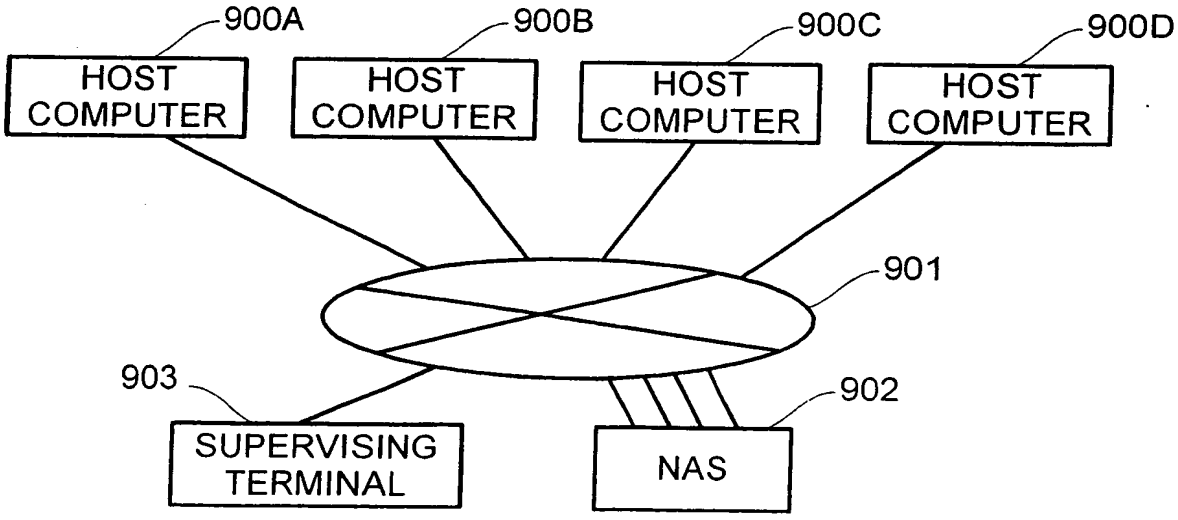
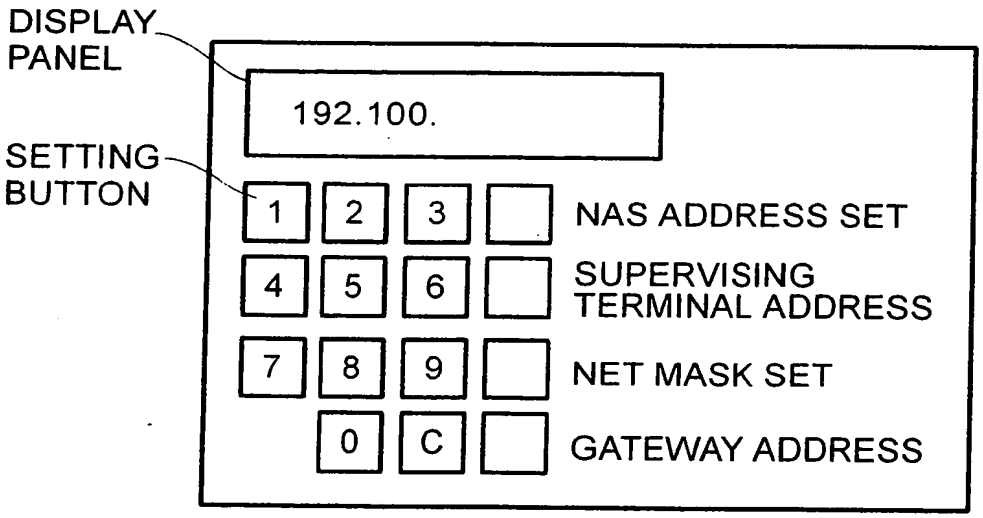

900A
900B
900C
900D
HOST COMPUTER
HOST COMPUTER
HOST COMPUTER
HOST COMPUTER
901
903
902
SUPERVISING TERMINAL
NAS

FIG.12


DISPLAY PANEL
192.100.
SETTING BUTTON
1
2
3
NAS ADDRESS SET
4
5
6
SUPERVISING TERMINAL ADDRESS
7
8
9
NET MASK SET
0
C
GATEWAY ADDRESS